Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Golf Swing Training Application

Harry Doyle, MCS

Supervisor: Dr. Kenneth Dawson-Howe

April 2023

A Dissertation submitted in partial fulfilment

of the requirements for the degree of

Integrated Master of Computer Science

Table of contents

Abstract 1
Table of contents 2
Report Overview 5
1. Introduction 6
1.1 Project Overview 6
Outlining the Scope 6
Setting our Aims 6
Explaining the Research 7
Our Motivation 7
Application overview 8
Results Overview 8
1.2 Overview of the Problem 8
Background Knowledge required 9
Understanding the Golf Swing 9
What is the Relevance of the Setup? 10
What is the Relevance of the Takeaway? 10
What is the Relevance of the Backswing? 10
What is the Relevance of the Downswing? 11
What is the Relevance of Impact? 11
What is the Relevance of the Follow-through? 11
2. Limitations and Challenges 11
Why the Speed of the Golf Swing is a Limitation 12
What is the Missing Frame? 13
3. Literature Review 14
3.1 Computer Vision in the Golfing Industry 14
3.2 Similar Applications 15
V1 Golf 15
Swing Index 15
Swing Profile 16
3.3 Academic Related Literature 16
Computer Vision based training applications 16
Computer Vision research in golf. 18
3.4 How will we determine if a swing was adequate? 19
Checks to Perform at Setup 20
Checks to Perform at Takeaway 21
Checks to Perform at Backswing 21
Checks to Perform at Downswing 22
Checks to Perform at Impact 23
Checks to Perform at Follow-through 23
4. Design Details 25

4.1 General Design Overview 25
4.2 Video Processing Methods Design Details 26
Video Preprocessing 26
Swing Stage Classification 26
Swing Analysis 27
MediaPipe Pose 27
4.3 Data processing Method Design Details 28
The Feedback System 28
4.4 Front-End Design Details 29
5. Implementation 31
5.1 Video processing implementation 31
Video Preprocessing Implementation 31
Swing Stage Classifier Implementation 32
Data Pre-processing 33
Stage Classification. 33
Swing Analysis Implementation 35
5.2 Implementation of the Feedback System 39
5.2.1 Data processing implementation 39
Feedback System Implementation 39
Processing for Solo Mistakes 40
Processing Root Cause mistakes 40
5.2.1 Front-end Integration 41
The homepage 41
The Analysis Page 43
6. Evaluation and Results 48
6.1 Gathering data 48
6.2 Evaluation of methods 48
Evaluation of the video preprocessing steps 49
Evaluation of the Swing-Stage Classification module. 50
Evaluation of Swing Analysis methods 52
6.3 Results 55
7. Conclusion 58
7.1 Further Studies 58
General Research 58
Research specific to our application 60
7.2 Our Closing Remarks 61

Report Overview

To help explain the process of answering the research question outlined by this project, the report will be split up accordingly to aid in developing an understanding of the problems we faced and the solution we implemented. The report is broken into 7 chapters highlighted below

- Chapter 1: Introduction - We will outline the scope, motivation and aims behind the study. We will also give the background knowledge required and finally give a brief overview of the entire project.
- Chapter 2: Limitations and Challenges - Here we will highlight the fundamental limits of the project with respect to the scope and aims. These limits will lay the foundations for the project.
- Chapter 3: Literature review - In this section, we will discuss previous research done in this field and also current uses of computer vision in the sporting industry. We will also explain the research we did in order to determine whether or not a swing was adequate.
- Chapter 4: Design - In this chapter we will consider the design of the golf training application. We also address any obstacles we expect to face in implementation and how we will overcome them.
- Chapter 5: Implementation - This section will explain how we implemented the various methods required to create such an application. These methods will be broken up into computer vision processing, feedback generation and front-end integration.
- Chapter 6: Evaluation and results - In this section we will outline the evaluation of any Computer Vision methods used in the implementation of a functional application. We will give details regarding how we could make possible further improvements. Once evaluating the methods we will discuss the results of the application as a whole.
- Chapter 7: Conclusion - In this section we will discuss any further studies that have come as a result of performing this research. These studies may come in the form of direct improvements to the application or studies that would have a larger more generalised scope. We will conclude with our closing thoughts and remarks.

1. Introduction

Computer vision is becoming widely used in various sports to provide augmented reality applications and statistic trackers. Computer vision has also been used to create computer-based training applications. This research hopes to expand the current range of computer vision based training applications in golf and aims to do so by facilitating an accessible method.

Scope of the study: Make the process of improving golf performance more accessible to amateurs.

Research Questions: By extracting key-features of a user's golfer swing via computer vision methods, can intuitive feedback be provided to facilitate easier improvements?

1.1 Project Overview

In this section we will further explain the functionality of our application with regard to the research question.

Outlining the Scope

This project aimed to develop a computer vision based Golf Swing coaching application to facilitate accessible improvement for beginners and amateurs. As such, the scope for our project is an accessible application that only requires a video camera from a user's device. By its very nature, we set out to design an application where a user could submit a video taken on their mobile device to a web-server. In the web-server we would perform low-power computer vision (keeping in mind in a production application, this processing would be limited to the process power of the mobile device) processes to analyse the video and compute the output.

Setting our Aims

The aim of this project is to create a fully functional, easy to use web application, whereby a user can upload a video of their golf swing. The video will be analysed by our algorithms to detect what is wrong with the inputted swing video (the checks we will perform will be outlined in the Literature Review). This analysis of the swing will be processed by the feedback mechanism, which will suggest a drill(s) to alleviate

the mistake(s) made. As the scope of the project outlines the target-audience as amateur and beginner golfers, therefore we must consider the user has a limited knowledge of the golf swing itself. As a result, the output feedback must tell the user what they are doing wrong, what this mistake may affect and yield a drill that can help alleviate this mistake.
This process can be executed on either one of the views of the golf swing (Face-On or Down-the-Line) or both if multiple videos are uploaded.

Explaining the Research

The research question we hope to answer is “By extracting key-features of a user's golfer swing via computer vision methods, can intuitive feedback be provided to facilitate easier improvements?”. To answer this question our application must perform computer vision techniques to extract the required data from the video to verify the required checks. To do so, we will use pose estimation. Then by checking which checks failed, intuitive feedback must be given. This intuitive feedback must help make improvements easier, and due to the assumed knowledge-level the user has about the golf-swing, the feedback will output a drill that will help fix the mistake. From this methodology, intuitive feedback to facilitate easier improvements is answered as the program will output the drill to improve and, as a result, the user does not need to do research into fixing the problem.

Our Motivation

The golfing world saw a major spike in popularity after being introduced to the 2016 Rio De Janeiro Olympic games. As a result of this new found popularity and throughout the Covid-19 pandemic the golfing world saw its biggest increase in popularity over the last 70 years. This was due to people being prohibited to take part in their preferred hobbies owing to the restrictions put in place to prevent the spread of the virus. As such, many people decided to partake in one of the few activities that was not completely outlawed for the duration of the pandemic, Golf. As an avid golfer myself, I understand the pain one must endure to reach the point of becoming good enough to truly enjoy the sport. As golf is a sport of fine proportions, to learn how to play is a time-consuming and expensive process. Current options to improve are either learning how to improve yourself from limited knowledge of what

results in better performance or paying a club professional to give a beginner lesson. Therefore, this research aims to provide an accessible method for new and intermediate golfers to improve without spending hours researching and reviewing swing features.

Application overview

To answer the research questions above, we designed an application that would take in a video of a golf swing as input. Using the acceleration change of the users wrists in the video we will parse the video to only include the golf swing itself. Once the video has been parsed, we extract a single frame of the key-stages of a golf swing. With each of these images we perform pose estimation on the image and verify the checks we need to do in order to determine if a swing was adequate or not. Once the swing has been analysed our program generates a readable object with the necessary information regarding the checks that we performed previously. Finally, this data is then processed by a feedback system to generate a drill that will alleviate any mistakes the user made in their swing. The drills recommended aim to create a swing capable of hitting the golf ball in the centre of the clubhead consistently.

Results Overview

In order to discuss the results of the application we had to analyse the various computer vision methods we implemented. Our evaluation requires more analysis, with more videos with varying types of swing and skill levels and a larger overall dataset. We established that our video preprocessing stage achieved an accuracy of 81%. Our performance for swing-stage classification was poor, only achieving an F score of 0.45. Finally our swing analysis module achieved an accuracy of 61%. Due to the low score in the swing stage classification, we ultimately failed to create an accessible application that will help users improve. However, we discuss improvements to our system that would likely result in an adequate implementation. As such our current implementation is more proof-of-concept.

1.2 Overview of the Problem

In this section we will discuss all required knowledge in order to fully understand how we aim to answer the research question presented to us. We will discuss the background of the project, including details regarding the golf swing and initial design choices. We will also highlight the key-features we are going to extract from the video and outline why we chose each feature and the impact it has on the resulting golf shot.

Background Knowledge required

In order to explain the checks that we will be executing in the application, we must first explain the breakdown of a golfer's swing and discuss the key components that influence consistency in a good swing, resulting in lower scores over time.

Understanding the Golf Swing

To answer the research question “By extracting key-features of a user's golfer swing via computer vision methods, can intuitive feedback be provided to facilitate easier improvements?“ we need to understand how these “key-features” impact the golfer's swing. How we selected these features will be discussed in the Literature Review, but we must first explain the basis over every golf shot, the swing.

A golfer’s swing is a movement that is repeated multiple times per hole. In order to excel in the game, a player must understand their swing enough to have an accurate estimate of the resulting shot. In essence, a golfer's swing should be consistent and not stray too far off its standard deviation, as they want to recreate the same shot every time. The golf swing is not a perfect science and there’s no one way to do it right[1], however there are numerous aspects that need to be considered in every swing. The swing is broken chronologically into 6 main phases: Setup, Takeaway, Backswing, Downswing, Impact and Follow-through each phase is as important as the other. The aim of a golf swing is to return the club to the right position at impact everytime[1], while generating as much power as you can. Power is generated by coiling your body in the backswing, and releasing the stored energy rapidly in the downswing to create as much speed as possible.

[Figure 1: An illustration of the stages of a golf swing[2]]

What is the Relevance of the Setup?

The Setup lays the foundation for the entire movement. In the setup a player will set the pivot of their swing, establishing the swing plane the club head will travel along throughout the movement. The golfer must also develop a stable stance that can accommodate the movement of the swing. A poor setup means you will need to compensate in the swing[1], leading to inconsistencies for the remainder of the swing.

What is the Relevance of the Takeaway?

The Takeaway is the initial movement of the clubhead away from the ball, in the takeaway the golfer should start the movement on the correct swing axis while aligning body rotation to the swing axis. A mistake here can have a significant impact as the movement axes being misaligned will have a linear impact as time progresses. A good takeaway will set the movement on the correct path to maintain balance whilst keeping swing plane and body rotation axes inline.

What is the Relevance of the Backswing?

In the Backswing a golfer performs the initial “coiling” rotation to generate as much energy as possible for the downswing. By properly executing the backswing a golfer makes the transition into rapid release of energy in the downswing much smoother, and so consistent. Stable rotation and movement is required in the Backswing to maintain axes alignment and balance established previously.

What is the Relevance of the Downswing?

For the majority of golf swings (i.e. full shots), in the Downswing the golfer begins the rapid release of the stored energy, to accelerate as fast as possible before impact with the ball. All while maintaining the swing plane axis to allow the club to make solid contact with the ball. Due to the majority of the movement occurring simultaneously and suddenly in this stage, this is where a golfer is most likely to make a mistake.

What is the Relevance of Impact?

Impact is the moment the clubhead makes contact with the ball and the ball starts accelerating forwards. It is the moment that you align your swing around to repeatedly strike the ball with the centre of the clubhead. This stage of your swing is the only part that directly affects the ball and any mistakes made previously will result in something going wrong at impact.

What is the Relevance of the Follow-through?

The Follow-through is the final phase of the golf swing, while it does not directly affect the outcome of the shot, it is a strong indicator that everything went as expected during the movement. Balance should have been maintained throughout the movement and the swing axes should only now start to lose alignment.

2. Limitations and Challenges

In this section we will discuss the limitations that we would need to consider in the implementation of such an application. These obstacles guide the design of the application as a whole, as they will lay the theoretical limits of the application, based on the project scope of an accessible tool (thus, hardware limitations such as camera specifications).

The main problems we face in the limitations of the application are 1) the speed of the golf club-head during the movement and 2) missing the frame of impact.

Why the Speed of the Golf Swing is a Limitation

[Figure 2: Distortions in the image due to high speed movement]

The biggest hurdle we will need to overcome in this section is the speed of the golf clubhead moving throughout the swing. In a golf swing the clubhead can reach a maximum speed of 120mph[<SITE but not reference????>]). Thus, in the case of a camera with a frame rate of 30FPS, the clubhead will move nearly 2 metres between some frames. As such the clubhead will be distorted in the video feed playback (amount of distortion will depend on camera shutter speed). This distortion will render typical methods of tracking objects useless such as template matching or Lucas-Kanade Optical Flow. Fundamentally, we will need to develop an algorithm that can track the high speed movement of a golf club's head, or an easier option, track the hands of the player as they move with much lower speed (approximately 25mph[<SITE but not reference????>]).

What is the Missing Frame?

[Figure 3: Movement between frames of impact]

As a result of the high speed movement of the clubhead, we may see the clubhead “jumping” roughly 2 metres between frames (depending on fps). While this does not introduce difficulty to the checks for many of the stages, it does introduce a problem when it comes to the checks we need to perform on impact. As the clubhead can move to such an extent, we will likely miss the point of impact and so we will not be able to carry out the various checks on impact. To compensate for this, we will track the features throughout the entire movement and extract the frame before and after the impact. We will be able to interpolate for an accurate estimate of the real value of the feature at time of impact. However, we may need to try various forms of interpolation to get the most accurate result.

3. Literature Review

In this section we will discuss relevant literature in the field, this will help to set our research apart from previous methods. We will discuss current computer vision functionality in the golfing industry and similar applications already in this space. We will also analyse previous research done in the topic, in both golf and other sports. Finally we will also explain the factors that create a good golf swing and what we will verify in the application.

3.1 Computer Vision in the Golfing Industry

In golf, the computer vision technology used is Trackman. Trackman is used in golf tournaments in order to trace the flight path of the golf ball after a shot. It uses a combination of cameras, radars and sensors in order to accurately track the golf ball. The image below illustrates the results of the technology.

[Figure 4: An example of the trackman technology used in golf TV broadcast[3]]

Recently the technology was expanded to track the clubhead and calculate information about the swing itself (i.e. the speed of the clubhead and swing plane axis). Thus, giving real-time information about the shot and the expected outcome. With this information Trackman is able to calculate shot statistics such as the balls distance travelled. Other statistics such as, ball distance travelled, launch angle and club head speed. The final implementation we must discuss are the Augmented

Reality models, in live broadcast we can see replays of the swings in Augmented Reality displayed in the studio. These replays give a 360° view of the golfer's swing.

In this section we have discussed various uses of computer vision in the sporting industry. These applications come from a need to create a fairer game, a safer game and even improve audience immersion.

3.2 Similar Applications

Here, we will discuss other applications that achieve a similar aim. In order to develop an adequate application we have to understand what the products already released and understand their methods. We will also explain how our implementation differs by automating the swing analysis. The 3 apps we will discuss are V1 Golf, Swing Index and Swing Profile.

V1 Golf

The V1 Golf app provides an interface for analysis of your golf swing. To do so, the app takes in an inputted video of your choice and will provide tools for analysis. These tools allow you to crop a video, compare and overlay it to another and draw various lines on top of the video. The video can then be submitted and analysed by a professional coach who will suggest various drills that can be practised to improve your golf game. The app comes with a database of hundreds of professional swings to compare against.

Swing Index

Swing Index offers similar functionality to V1 Golf but in a very different method. In this application a user uploads videos of their golf swing, which is analysed and marked by a coach. Once it has been marked, the results are passed through an AI model which will rank your average swing and produce a personal learning roadmap in order to improve your game.

Swing Profile

Swing Profile is an app that allows you to record your golf swing. The app will then classify the key-frames, allowing the user to go through each stage image and verify

that their swing occurred as expected. The app allows a user to verify that their swing is on the right axis at all times throughout the movement and also allows you to compare your swing against other videos. However the app fails to provide any useful feedback or features that allow you to check other key factors of a golf swing.

All these applications have one thing in common, if you want to improve your golf game you need to either analyse your swing in detail by yourself or a coach must do it manually. Our application aims to mitigate this and aims to provide feedback to the user based on what has occurred in their swing and how they can improve it without the need for human intervention.

3.3 Academic Related Literature

In the previous section we talked about current implementations of computer vision in the sporting and golf industry. In this section we will discuss academic research in computer vision with similar aims as we have for this project.

Computer Vision based training applications

This section will discuss previous studies in the area of computer vision based training application. We will highlight previous studies in creating an application that analyses an athletes movement in order to provide insight into what could be done differently in the field and have shown a true use across various sports.

While the first item we will discuss here isn't a research paper but rather Apple's developer tutorial for using their computer vision methods to implement an application. The tutorial provides instruction into building a feature-rich app for sports analysis[3]. In the tutorial they outline how a developer can use Apple’s provided pose estimation model in order to publish an app designed for improving Bean-Bag tossing performance. The tutorial demonstrates how pose estimation and computer vision can be used to analyse a video of a sport related movement and provide feedback to improve performance. This highlights Apple's belief that this technology can be used in designing intuitive applications that aim to improve sporting performance, have invested in the technology and promote its use by developers to design similar applications for various different usages.

The first paper we will discuss is titled “trAIner - An AI Fitness Coach Solution"[5] and was published in 2022. The paper aimed to implement an automated fitness coach solution which performs all the tasks of a physical personal trainer[5]. The paper outlines the authors research into computer vision (more specifically pose estimation) and machine learning to analyse an inputted workout video. Data is then provided to give feedback to the user in order to correct their movement and avoid injury (based on the user's inputted health data). To do so, the application takes in a real-time video feed and runs MediaPipe pose estimation on the video to generate x,y and z coordinates. Furthermore, a Random Forest Classifier model is run to find the end of the movement. During the movement the pose estimation points are analysed in real-time to provide feedback if the movement occurred correctly. If the probability of the movement being performed correctly isn't above a threshold (80%), auditory feedback is given in order to advise the user to correct their form. The research expanded on previous research[6] in the area which did not provide real-time information about the exercise performance. The paper failed to provide results for the implementation of the application, however, the research is still relevant as it describes an implementation of a computer vision based training application. The core of the application runs on pose estimation which our research will also use as a core technology.

Another study analyses the use of pose estimation in a Virtual Self-Defence Trainer[5] application. The paper outlines their use of OpenPose pose estimation to analyse and score user performance in self-defence. However, this paper fails to provide the user with feedback that will ultimately help their performance, but rather relies on the user understanding their feedback enough to self-improve by being shown what is flawed in their movement. The author outlines how data extracted from the pose estimation algorithm is processed using various data analytic methods. This processing yields a score that can then be compared against other video sequences. While the paper uses pose estimation to extract information about the user, the majority of the processing is data driven and lacks any other computer vision based functionality. However, it is mentioned in the further research that the system could be expanded to include more movements which would require heavier computer vision analysis.

The final paper we will discuss implements “A Trainer System for Air Rifle/Pistol Shooting”[7]. This paper aims to develop an application that can replace a trainer in a professional athletic environment, rather than one used for casual use. The paper outlines research into how they designed an application for the Sri Lankan Olympics team to help them win a gold medal. However, rather this application is designed not to replace the trainers themselves, but rather “assisting the coaches’ decisions”. The authors designed an application which uses various sensors and cameras to analyse the performance of the athletes in a low-cost manner. This paper shows that an adequate implementation of a computer vision based sports trainer application could be used in order to aid professional athlete training regimes in a competition with the world's-best in their respective sports.

In this section we have outlined past research in the area of computer vision based sport training applications. We described such applications in two different sports, with two different main end-goals. We highlighted that pose estimation is a viable method of improving athletic performance in terms of both safety and skill.

Computer Vision research in golf.

As we previously discussed, past research in the sporting industry aims to bring computer vision into a training regime for athletes. In this section we will discuss golf-oriented studies that deal with problems we expect to run into as our application is developed.

The first paper we want to discuss outlines the key-factors in Rotation Biomechanics of the Elite Golf Swing[8] that result in optimal power generation. The paper studied the factors that most affect the speed of a golf swing and established that the key-movement in creating power is in the backswing. This is due to the rotation plane that the movement will occur on and it is the initial winding up of the “spring” to release at impact. In the creation of our application we should consider that the main factor of distance in golf is related to club head speed at impact. As such, our training application should only suggest changes that will not alter movement speed.

Other studies have been done in order to extract the key stages of a golf swing from an inputted video. This study[9] outlines an application that will extract key-frames of the golf swing in order to provide further analysis. Logic used in this paper will be used in the development of the application as we will need to extract the key-frames from the video in order to analyse the players pose to gather data so as to perform the necessary computation.

The final paper we will discuss adopts a similar method as us, however, uses machine learning to compute complex patterns in the movement. The paper uses Pose Estimation to predict performance using Edge processing[10]. This paper implements a solution using video data converted into points via pose estimation. These points are then preprocessed before being used to find critical points in the swing. These points are analysed using various methods, however, the computation is more data-driven rather than video-driven. Unlike this research, our project uses computer vision as the analysis tool. In contrast, our application is designed with the intention of using computer vision techniques to analyse a given video of a user's swing and analyse it in the same manner as a coach.

In this section, we discussed studies used to design applications using pose estimation to provide a trainer application in various sports including golf. We also discussed previous research in the golfing field which varies from a trainer based application to solutions to problems we expect to face. We also highlighted the differences between our proposed application and previous implementations that have either been studied or released for general use.

3.4 How will we determine if a swing was adequate?

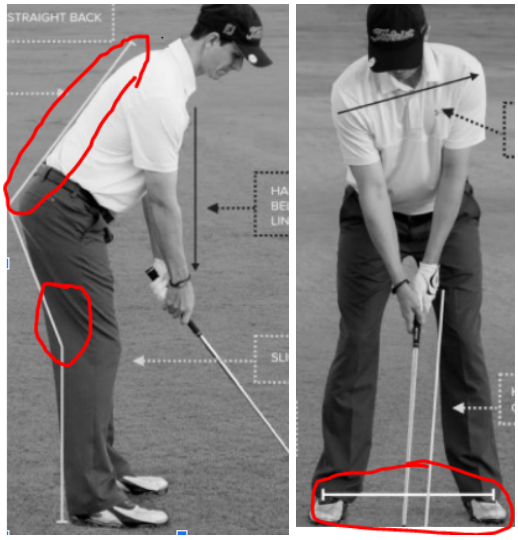
This section outlines the relevant research we did in order to find the crucial factors we would need to check in order to determine if a swing was correct. An important thing to consider in the development of this application is that the existence of one 'perfect' golf swing is a fallacy[11]. In order to design an application that will ultimately benefit the user, ideally we would create an application that is tailored for every

individual golfer and their swing. Instead of designing a personalised analysis application we aimed to verify that a number of important fundamentals of the golf swing should be adhered to, in order to achieve the best regular and long term improvement[12]. Studies have shown that the average PGA professional has a standard deviation of less than that of half of an intermediate golfer[12]. The key to improving your golf game is more consistency between your shots and so we wanted to design our training application to verify that key factors in creating consistency were adhered to in a swing. All these key factors we will discuss are found in citation 12 as these lay the foundation for a good golf swing and should be adhered to as these are the “fundamental parts of a sound golf swing”[12].

As the golf swing is a simultaneous rotation of the whole body, it has a lot of moving parts which can often lead to minor mistakes which can drastically change the course of the swing. These basic rules help to steady the axes the swing uses as its foundation, keeping the clubhead square and maintaining balance throughout the quick movement. Thus, as there is less movement in the foundations of the golf swing, we can help regularise the swing of the user in order to reduce the standard deviation between swings. All these features can be extracted using 2 camera views (from a “down the line” view and a “face on” view) and various computer vision methods.

Checks to Perform at Setup

In the set-up the player creates the initial foundation of the swing to reduce the fluctuations in swing-movement. A straight back enables the player to rotate around an appropriate axis, thus, if we can indicate that the back is not straight we can inform the user who can make adjustments in order to create a straight back and a more similar swing. A slight-knee flex and the feet being shoulder width apart will help balance the user, thus reducing the number of moving parts and keeping the rotation axes stable, improving the standard deviation between swings.



[Figure 5: Displaying the checks we must perform in the setup]

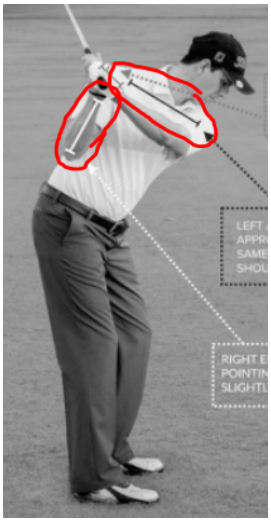
Checks to Perform at Takeaway

In the takeaway the entire rotation should move simultaneously to synchronise the rotation axis creating a more consistent movement. The lead arm should be slightly bent, mitigating wrist rotation that can cause inaccuracies. Finally, the shoulders and arms should move together as a triangle to result in one piece movement, keeping the rotation axes synchronised.

[Figure 6: Displaying the checks we must perform at takeaway]

Checks to Perform at Backswing

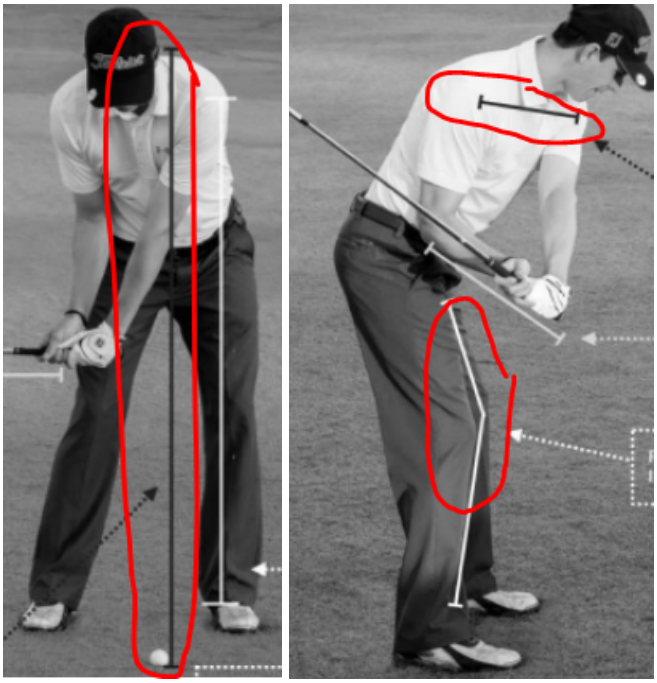
If the backswing is consistent in remaining on the appropriate rotation axis, the shift to the downswing will be smoother. The backswing is where the rotations begin, in the swing the rotations are almost elastic whereby we must “wind up” rotations in order to release them at the appropriate time in the downswing. In the backswing we must ensure the left arm is approximately on the same plane line as the shoulders. This is the first marker that our swing is on the right axis for downswing and impact. The right elbow must also be pointing down and slightly left to ensure a more consistent swing. These features verify that we are ready for the downswing and to generate the power for impact, while keeping our clubhead square to the ball.



[Figure 7: Displaying the checks we must perform in the backswing]

Checks to Perform at Downswing

In the downswing we must ensure that the user’s shoulders are slightly closed to gain balance and reduce the likelihood of moving off axis. The flex established in the legs in the setup should be maintained to this point in order to maintain swing plane and power, thus recreating a consistent swing. The player's head should also be positioned slightly behind the ball, this prepares the user for solid contact with more power.



[Figure 8: Displaying the checks we must perform in the downswing]

Checks to Perform at Impact

On impact we need to ensure the hips are slightly pointed upwards to ensure that maximum power has been delivered to the ball for optimal distance. We also need to ensure that the lead foot, leg and shoulder are inline to ensure maximum speed as a result of the rotation on many axes simultaneously.

[Figure 9: Displaying the checks we must perform at impact]

Checks to Perform at Follow-through

The follow-through is the final phase of the golf swing. While the follow-through does not directly affect the shot, it is a strong indicator that our swing stays on the appropriate rotation axes and balance is maintained throughout the swing. In the follow-through to validate these the user's right foot should be on the toes, this will highlight that balance and weight shift has been appropriately distributed throughout the movement. We also must verify that the body is facing the target, indicating that the rotation was executed correctly and finally the head, right shoulder, hips, left knee and left foot are all in one line, indicating balance and weight distribution were maintained.

[Figure 10: Displaying the checks we must perform in the followthrough]

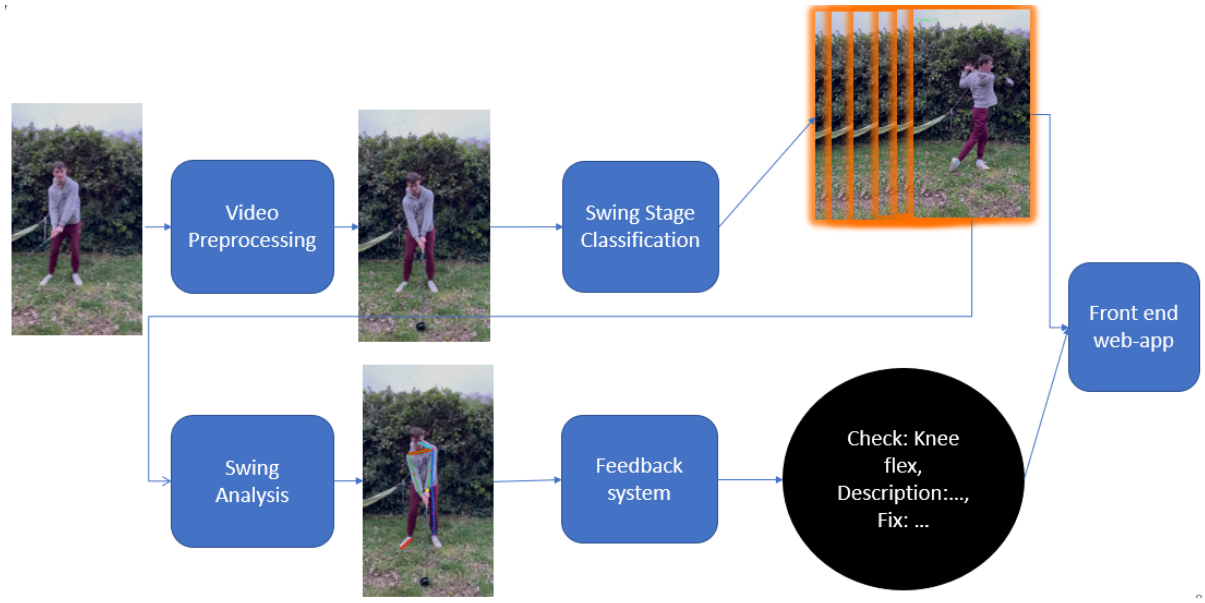
This section discussed our research into how we will validate if a golf swing occurred as expected, following all the fundamental rules that lead to consistency. From the images inserted in the sections above we can see that the majority of these features can be tracked by implementing a pose estimation algorithm that will identify all of the required key-points. By implementing checks in the program to verify these fundamental rules are adhered to throughout the swing movement, aiding beginner and intermediate golfers.

4. Design Details

The previous sections outlined the desired functionality of the application and set the boundaries for our application. This section will discuss how we designed the application around the aims above, providing insight into why we decided to design the functionality.

4.1 General Design Overview

In order to create the desired application, we need to consider how to analyse the swing, perform the checks and then provide output. In the section we will discuss how we designed the process around these tasks.



[Figure 11: A snippet from my presentation of an animated slide that illustrates the data flow of the processing]

The first design decision we made was to isolate the Golf-Swing movement alone from the video inputted by the user. The video uploaded by the user is likely to contain irrelevant frames, as such the first step in the process will be to preprocess the video data and parse all irrelevant frames. The resulting output will be a video that consists of only the swing itself.

The next step in the process is to prepare the inputted data for the analysis. The checks described in the Literature Review do not require continuous analysis of the video, rather they require these checks being performed at a certain stage of the golf swing. As such, the parsed video must be classified into the individual swing stages

that the checks must occur in. These classified images must then be inputted into our swing analysis methods. Here is where we will analyse and extract the “key-features” required to perform the checks.
The data from the Swing Analysis would then be sent into the feedback system, where any item that was marked as a mistake is processed to yield a drill that will alleviate that mistake.

4.2 Video Processing Methods Design Details

From the graphic above we can see the first 3 stages of the process all use video or images to process data. In this section we will discuss the classes that are centred around computer vision methods.

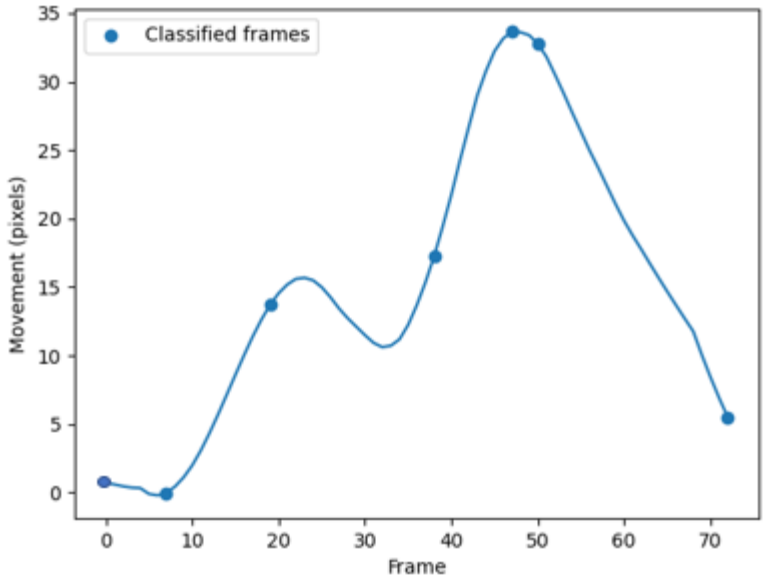
Video Preprocessing

In order to parse the user's uploaded video to isolate the golf swing, we must consider the movement of the golf swing. For the duration of a golf swing a golfer aims to achieve the highest possible speed that the clubhead can move for impact with the ball. However, we are not tracking the clubhead as it is distorted in the frame by frame feed of the video. Instead, we can use similar logic and track the players wrists as they are subject to major movement in the golf swing. Once the point of highest speed is identified, we can parse a static number of frames either side and so parse the video to isolate the swing.
This works in the expected videos as the “irrelevant” frames that are expected to be in the video, contain very little wrist movement.

Swing Stage Classification

In order to perform the necessary checks, we must have single images of the golf-swing stages: Setup, Takeaway, Backswing, Downswing, Impact and Follow-through. As we know the goal of the golf-swing is to create as much speed as possible at impact, while maintaining consistency. This results in the swing stages logically following an order with regard to time and speed. The setup is going to contain no movement as the golfer should remain stationary as they establish the desired stance. The takeaway will contain the initial movement as the golfer starts aligning their wrists with the swing-plane. After the takeaway the user’s wrists are

going to accelerate as the movement progresses to the backswing. At the pinnacle of the backswing we will see the acceleration rapidly stop as the golfer prepares to shift the backward-movement into the forward-movement of the downswing. In the downswing we are going to see a rapid increase of the acceleration to the point of highest movement (impact). Finally, after the point of highest speed, we are going to see the acceleration gradually decrease as the flowthrough progresses. The acceleration curve is graphed below.



[Figure 12: This graph displays the acceleration curve of the users hands as time progresses]

Swing Analysis

From the checks described previously we can see these can be verified by analysing various limbs on the body. The points we need to track vary depending on view, but contain parts of the hands, torso, legs and head. As such, Pose-Estimation was selected to implement the analysis. We decided to use Google's Mediapipe Pose Model.

MediaPipe Pose

The pose-estimation model is equipped with everything we need to perform the various checks and is available on iOS, Android and Python. MediaPipe Pose operates on the BlazePose architecture to offer processing with low-complexity

computation, this is important as our scope outlines an accessible method of improvement and so, we need to assume low-power devices. Blaze pose is uniquely suited for fitness applications[4]. BlazePose operates on a COCO topology, which is modelled by a “Bottom-Up” approach. A “Bottom-up” model will locate the necessary pose-landmarks taking the entire scene in as input (instead of segmenting the person first). In BlazePose this is done by generating a heatmap which corresponds to the probability that a pixel belongs to a body part. This heatmap is then processed to generate the pose-landmarks.

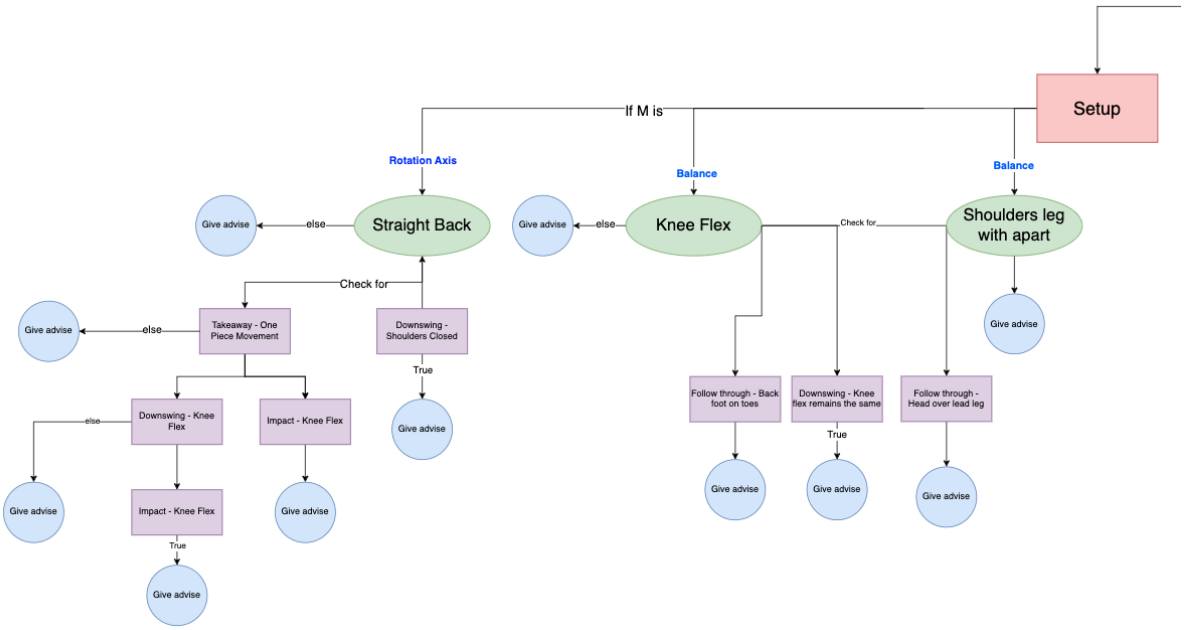
Using the MediaPipe pose model we will track the necessary body-parts in order to perform the checks needed. To do so, we will extract the 2D [x,y] coordinates from the classified image stages, then verifying the checks required. After analysis the program will output data containing elements for the check that occurred.

4.3 Data processing Method Design Details

The above section explained how the processing will occur when the data being processed has a visual element. However, swing analysis marks the end of such processing as it fills in readable data containing details of the analysis. In this section we will discuss how we process this data.

The Feedback System

The feedback system is going to process the mistakes identified and suggest a drill when a mistake has been made. This feedback system will follow a tree-structure and each mistake will be processed down the tree. However due to the nature of the golf-swing, a mistake made earlier in the swing may be a root cause of one identified later in the swing. By its very nature, the feedback system must be robust to this use-case and must only yield to drill to fix the root cause of the error. Relevant information regarding mistake characteristics (such as root cause and leads to other mistakes) is stored in the metadata.



[Figure 13: Sample section of the architecture for the feedback system. For expanded view the file can be located here]

4.4 Front-End Design Details

The project description states that the program should be accessible and easy to use. As such, we decided to design a simple web-application which would facilitate calling the algorithms provided by this project. This front-end application would be easy to use with clear indication of mistakes made and how to correct them.

The application will require the user to provide video input and so, the front end will need to provide functionality to allow the user to upload the video they want to assess. This upload functionality should only be called once and should temporarily save the user's submitted video to the web-server.

The system should be capable of displaying this data in an easy-to-use manner, all relevant information being displayed by a user interaction from within the application. The front end should display the checks that were analysed regardless of if a mistake was made or not, as such we should be able to distinguish easily between a check that failed or succeeded. The main analysis page will display the 6 classified swing stages as modal style windows. When a classified image is opened the front-end will render buttons for each check that was analysed in that image. The

checks that failed will be made obvious by setting the background colour of the button from green to red. When the check has been selected the front-end should use the data gathered in the swing analysis and feedback stages to draw over the image. The data that will need to be displayed to “facilitate easier improvements” is the check title, the points required to make the check, description of the mistake and what it can affect and finally a drill to help alleviate the mistake.

5. Implementation

In this chapter we will discuss the implementation of the program with respect to design. We will outline any issues we encounter and how we overcome these obstacles. As previously mentioned, the processing is sectioned into 3 sub-topics; video processing, data processing and finally front-end integration. We will maintain this structure in this section as this follows the flow of data.

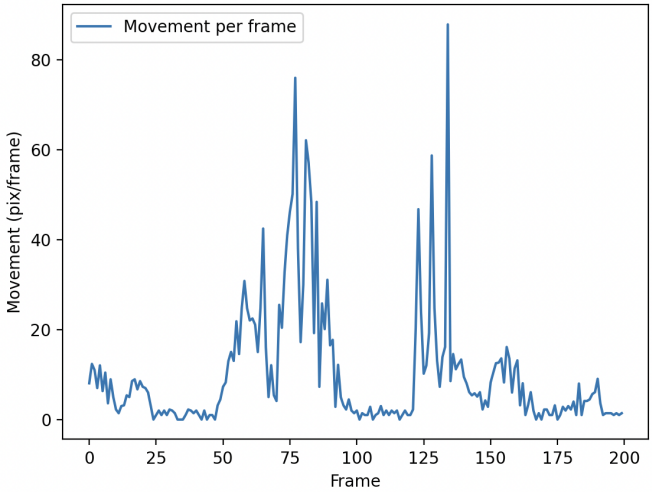
5.1 Video processing implementation

In this section we will explain how we implemented the modules that use visual data as their input. The aim of these modules is to analyse the swing data provided by the user and verify if the checks in the Literature Review were adhered to.

Video Preprocessing Implementation

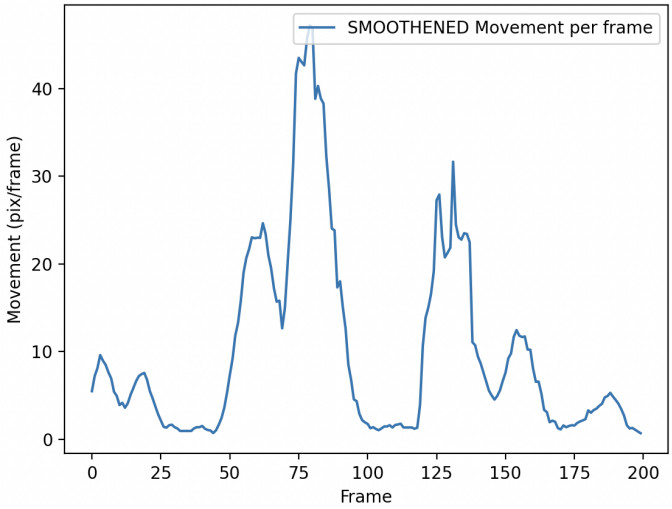
To reduce computation in analysis we implemented video pre-processing. The process of preprocessing the data in this project was relatively simple by using the point of highest movement of the users wrists and parsing the video around that point.

To implement this functionality we had to analyse the speed of the users wrists through the process of the video. To do so, we must first play an initial run of the video feed to record this data. In this run through we will record and track the users wrist as image-plane coordinates (i.e. [x, y]). Once these results have been recorded, we calculate the distance the point moved between frames. Resulting in the data below.



[Figure 14: Displaying the recorded movement of the user's wrists in the full video]

From the data above we can see the logical "speed" of a golf swing (between frame 50 and 100) that we have previously discussed. To reduce errors we smoothen the data by convolving it against a normalised 1D kernel, which will get rid of high-frequency noise. In this calculation, the dot product of the kernel and data is calculated and so smoothing the curve by reducing the effect of outlier data via weights. Note: in the graph above the quickest movement in the video is actually at frame 145, but this frame is not actually in the swing of the video but rather movement after the swing. Applying convolution to this data lowers the significance of this movement as within the kernel window the weighting is far less than that of frame 75, which is the fastest point in the golf swing. The graph below shows the result of this smoothening.



[Figure 15: The recorded data after applying convolution]

From this point we can now parse the video around this data, as mentioned we know the point of highest acceleration is the impact frame which is roughly halfway through the golf swing. We then parse the video to isolate the golf swing by defining a set number of frames either side of this point of highest acceleration.

Swing Stage Classifier Implementation

In this section we will discuss how we extracted the single frames for the separate stages of the swing as we needed these to verify the checks. As mentioned in the design section, while every golf swing is unique, there is a pattern in the speed. While the method above also used the speed to preprocess the video, now that we will be following logical patterns we will be far less robust to miscalculations and noise. As a result of this, the processing here is different to that above.

Data Pre-processing

The first step in the process is once again to track the wrists of the user in the video feed and record the 2D image plane coordinates. Calculating the movement from this alone however is not enough as there will be some error in the location when the pose estimation will miscalculate (i.e. occlusion). In order to avoid this, we first remove outliers in the data and estimate their real location.

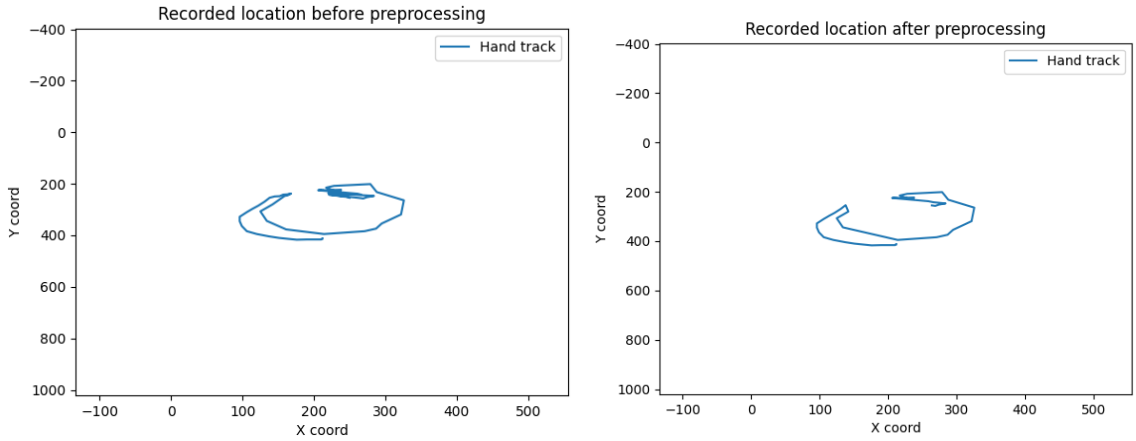
Outlier Removal

To remove outliers in the data we use Z-Score thresholding. This works by calculating the z-scores for the 2D coordinate data by calculating the z-score for both axes individually. These values can then be tested against a threshold, any point greater than the threshold value stray too far from the standard deviation and so are outliers. The corresponding data is then replaced by None values.

Outlier Estimation

In the new data we are left with None values where coordinates should exist. In order to fill in these values with realistic estimates, we need to examine the data further. Our data contains accurate calculations of the location and None values for where a value is not realistic. We also know that our index for location corresponds

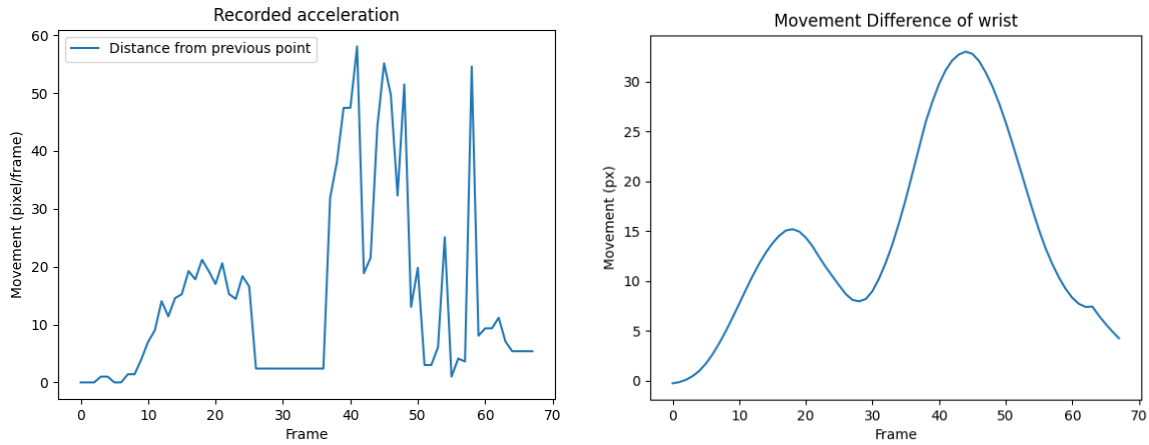
to the frame index of the video which will play a key rule as a variable for time. As the user's hands move in a rotation with respect to time, we can estimate the missing points by fitting a line between the last known location and next known location. This line is then used to calculate the estimate by splitting it up into equal sized segments, where the number of segments corresponds to the number of None items between the 2 known points. In this way this method can calculate estimates regardless of the amount of consecutive miscalculations. The graphs below display the results (left unprocessed, right processed)



[Figure 16: Displaying the difference between the unprocessed and processed data. Note: Most significant effect of this processing is around (250, 200) coordinates]

Stage Classification.

Now that our data has been preprocessed to remove and re-calculate outliers we need to calculate the movement changes of this data as the swing progresses. This has been made simpler by our previous video preprocessing, as the video only contains the swing, this allows us to easily process the data. However, as the methodology we are using for classification is sensitive to even minor errors, we need to first smoothen the acceleration curve. This is done using a Savitzky-Golay filter. The Savitzky-Golay filter will calculate a kernel of window-length containing the weighting of the of the points in the window and fit a polynomial function to this data, calculating the midpoint (as such kernel length should be an odd number to allocate a spot for actual midpoint) of the data. The resulting data (below) will be smoothened to adhere to the methodology of the swing speed.



[Figure 17: Showing the before and after smoothening of the acceleration curve using a Savitzky-Golay Filter]

As we can see from the smoothened curves above, our data follows a curve with no irrelevant peaks or dips. As such, to classify the stages using the methodology that the speed of a golf swing changes depending on time and stage we can travel along the curve. As our index variable corresponds to time and our swing stages occur sequentially, we know the first image to classify is the setup. In the setup we know the golfer is going to be still and so we can travel along the curve until the first local minima. Once this data has been calculated we get the index of the apex and save that index from the video (Note: while the speed of the movement is between 2 frames, we use an offset of +1 to make sure we use the frame after the movement has occurred).

The next frame to classify is the takeaway, the takeaway is at the beginning of the backswing movement. The next local maximum is the point where the backswing is fastest. By this logic, with a line from the setup frame to the peak of movement in the backswing can calculate an estimate for the index of a point 1/5th of the way up the line, this will be the frame of the takeaway.

The frame marked as the backswing is going to occur at the end the backswing, as momentum shifts the acceleration is going to progress positive to negative. As such, our speed graph above is going to experience a dip (but not stop due to our smoothening) at the end of the backswing movement. To get the index of this frame, we simply travel from the previous local maximum (highest speed in the act of backswing) down the slope until the curve trends upwards again.

The next frame to classify is the downswing. Again the act of the downswing is continuous but the frame we need to classify is at some approximate point along the increase of speed. Similarly to classifying the takeaway, we can connect a line from the backswing data to the point of highest movement (which will be the impact). As we want the downswing frame towards the end of the act of the downswing, we can calculate a point ¾ of the way up this line and save the approximate index of this point.

Classifying the frame for impact is actually more complicated than one would initially think. As mentioned in the Limitations and Challenges section, the swing movement may be too quick to catch the exact frame of impact. Due to this, we need to classify the frame before and after impact (we will discuss why in the Swing Analysis Implementation section). To do so is simple, we can get the maximum value of the movement and record this as pre-impact and the following frame as post-impact.
The final stage we need to classify is the follow through, once again the follow through is a continuous movement after impact, however for the checks we must perform we want the frame at the end where there is no movement. As a result of this we can simply traverse from the points of highest movement down the curve, until the data trends upwards and save the frame at that index.

Swing Analysis Implementation

The previous section discussed how we classified the images for each swing-stage. These classified images are used as input into our swing analysis module. In this section we will discuss how we implemented the functionality for executing the checks and generating data in a meaningful manner.

The first thing to consider for the swing analysis module was to eliminate the video as the main processing data and record the analysis in a manner that the user can understand. As such, the aim of this module is to analyse the swing-stage images and generate informative data that can be understood by the user. The data output from this module will be structured in a JSON object, with the key-value pairs representing information regarding each check.

In order to perform the necessary checks, we implement MediaPipe pose estimation to locate the relevant pose-landmarks. We use configuration parameters to set a minimum tracking confidence of 75%. Setting this parameter will eliminate any predictions that are blatantly incorrect.

We must first select all relevant landmarks required for the checks we are performing, in order to do so we can select a subset of features returned by MediaPipe that includes all relevant points that we need for our processing. This is possible as MediaPipe returns the landmarks in an array where the index corresponds to a different point on the user's body. MediaPipe outputs the predictions of all pose landmarks as an array of objects, we extract the [x, y] coordinates of the landmark. However, as MediaPipe calculates these as normalised coordinates we must first calculate the coordinates of the landmark as OpenCV image pixel coordinates in order to properly visualise them. To do so we follow the below formula.

$$[p_x, p_y] = [round(n_x * I_W), round(n_y * I_h)]$$

$where$:

p_x $is\ pixel\ coordinate\ "x"\ value$

p_y $is\ pixel\ coordinate\ "y"\ value$

n_x $is\ normalised\ coordinate\ "x"\ value$

n_y $is\ normalised\ coordinate\ "y"\ value$

I_w $is\ the\ input\ image\ width$

I_H $is\ the\ input\ image\ height$

$round()\ is\ a\ function\ to\ round\ to\ nearest\ whole\ number$

Now that we have calculated the subset of points we need and assign them into a new array where the index represents a body part location in image coordinates for OpenCV.

With our pose estimation implemented, we can now use the points found to execute the analysis in order to generate data about the user's swing. In this section we will describe how we performed the checks and generated the data, with reference to each check we can process. In this explanation we assume we are processing when

input of both views were provided (but the program also offers functionality of single view processing).

As mentioned previously we have a JSON object as an output of this stage, this object follows the below template.

```
data = {
    "Check": <Name of Check>,
    "Stage": <Swing stage to check>,
    "Problem": <Imapct on swing>,
    "Description": <Description of analysis>,
    "Fix": <A fix/drill generated from the feedback>,
    "Points": <Array of required coordinates>,

    # Remainder used as metadata
    "isMistake": <Did the check fail>,
    "isRootCause": <Does this mistake lead to another>,
    "LeadsTo": <If so, list of all the checks it can affect>,
    "isProcessed": <Has this check been processed by feedback>
}
```

[Figure 18: Displaying a template of the analysis output JSON object]

The fields in this object are all used for various reasons in the remainder of the processing, some are used to provide understandable data and some are used as metadata for processing. For each check that is performed, its related data is appended to an array which is used in the feedback generation and frontend visualisation. Some of the fields here are static and so are hardcoded in each check, in analysis the only fields that are changed are “Description”, “isMistake” and “Points”. The “isMistake” field is marked as true, if a mistake was identified in the check that occurred. The “Points” field will be filled in regardless of if a mistake was made or not, containing the image coordinate values of the points required to perform the check, this is done in order to allow easier visualisation of the check on the front-end. The “Description” field is filled in with different data depending on if a mistake was made and how severe it was.

In the setup we need to verify that the legs are shoulder width apart and that a knee flex capable of keeping balance is established. To check if legs are shoulder width apart, we must run the pose estimation class on the setup image from the Face-On view and get the coordinates of both shoulders and both feet. We can then verify that the X coordinate of the foot is roughly equal to that of the corresponding shoulder. If

a mistake has been identified the “isMistake” field and the relevant readable data is filled in the “Description” field i.e. “Your right foot is too far ahead of your right shoulder”. Next we must run the knee flex analysis on the Down-the-Line view and acquire the coordinates for the trail leg ankle, knee and hip. We calculate the angle formed between the users ankle, knee and hip. Filling the “Description” field with information based on the angle formed and verifying that the knee is directly above the ankle and “isMistake" if a mistake was identified.

The checks we must perform in the takeaway are to verify one-piece movement and that the trail arm is straight. To verify one piece movement we must check the elbow for both arms is kept straight during this movement, to do so we get the coordinates of the shoulders, elbows and wrists and calculate the angle formed between points and respective elbows. Verifying that the trail arm is straight occurs in the exact same way but only on using data calculated for the user's trial arm from the Down-the-Line view. The relevant data is filled in as necessary.

In the backswing we need to verify that the user's lead arm is on the same plane as their shoulders. To check this we must get the coordinates of the users shoulders and calculate the slope of the line connecting these. Checking if this slope is the same as that formed by the line connecting the user's lead wrist to their lead elbow. The relevant data is filled in. We also need to check that the user's trail elbow is pointing down and to the left. This is done by calculating the angle formed between the wrist, a point directly under the wrist and the elbow. Then we verify that the angle is in a range greater than 10° and 30°, thus indicating that the elbow is pointing both down and slightly left.

In the downswing we are required to verify that the users shoulders are closed (pointing to the left of the target, as such we can verify that the slope of the line is between 0.1 and 0.3 (as less than 0.1 will indicate too far closed and greater than 0.3 will indicate open). We also need to verify that the knee flex angle established in setup is maintained, this is done by checking the value of the angle that was calculated is approximately equal to that in the downswing (not exactly equal as there will be minor differences in location). Performing these checks results in the below images.

The frame of impact is the most difficult place to perform the checks, this is a result of the high speed movement and the exact frame for impact being missed. As a result of this we must interpolate between the value before and after impact. This can be done as our input video is stationary (i.e. the camera is in the same place) and so we can approximate the values at impact. This is done by calculating the coordinates of the relevant points in the frame before and after impact and calculating a point in the middle of this. These values correspond to the point directly in the middle of these 2 frames, however realistically impact can happen at any time between these 2 frames. As such, this method is not perfect, but gives a very close representation of the real values.

In the follow through we must only verify that the golfer's head is over their lead leg. To do so we simply get the coordinates of each body part and check that the X coordinates are roughly equal to each other.

The resulting output of the Swing Analysis stage is a list of JSON objects, containing all relevant details about all the checks that were processed. The data will now be processed in this form instead of as a video.

5.2 Implementation of the Feedback System

5.2.1 Data processing implementation

In the previous section we discussed how we generate readable data from the analysis of the users golf swing. In this section we will discuss how we further process this data and generate feedback for the user to practise.

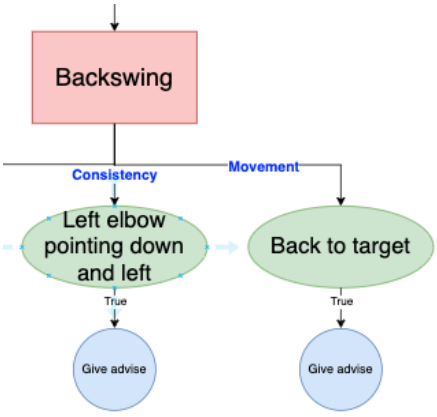
Feedback System Implementation

Once the user's swing has been analysed, we have identified the mistakes the user made which may affect the outcome of their shot. In this section we will discuss how we use the analysed checks to find a drill that can help alleviate the mistakes the user made.

The feedback system follows a tree structure with hardcoded outcomes for various scenarios. The output data from the video analysis stage is a list of all the checks that occurred and data about the checks. As we only need to generate drills for mistakes that were identified, the first step in implementing this module is to filter the array to only contain elements where the “isMistake” field has been marked as true. The mistakes made come in 2 varieties: solo mistakes and root causes.

Processing for Solo Mistakes

A solo mistake is any mistake that does not lead to another flaw occurring later on in the swing. As such, the solo mistake is only called when the mistake being processed is not identified as a “rootCause” in its metadata. The first step in order to process the mistake is to check which swing stage the mistake occurred in. To generate a drill we use the data in the “Description” field which will tell us more about what happened in the swing. If the check that failed is not a “binary” mistake (i.e. there is a measurement to how incorrect it was) this will be indicated in the “Description” field and so the various cases that could occur will yield a different drill. The final step in the process is to mark the “isProcessed” field in the metadata as true, this will avoid regenerating a drill in the recursion steps.

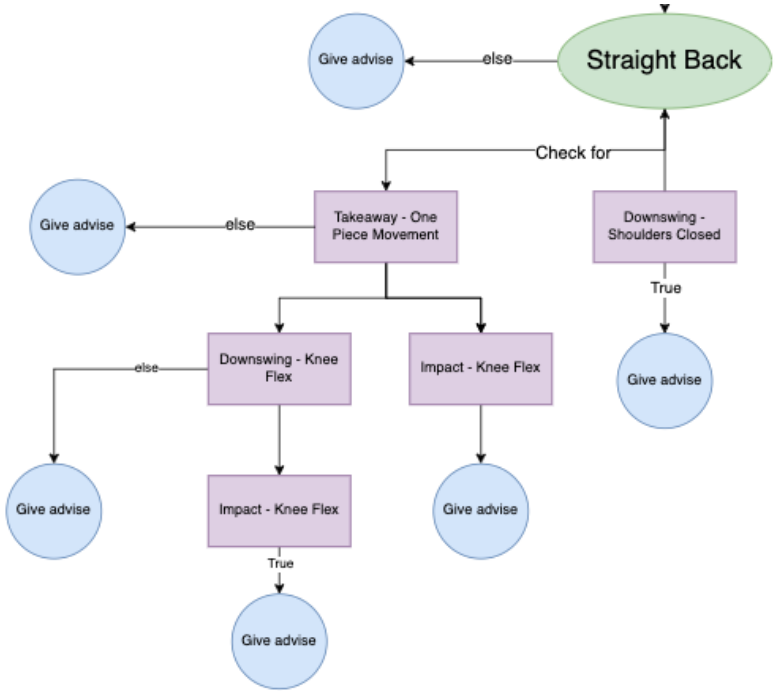


[Figure 19: Solo mistake node from the feedback architecture tree]

Processing Root Cause mistakes

The root cause mistakes are mistakes that may lead to other mistakes occurring later on in the swing. An example of a root cause mistake is if the knee flex in the setup is not correct, this will mean balance will be affected for the duration of the movement. As a result, the signs of balance being maintained throughout the swing

(such as the head over lead leg check) are likely to also fail. As such these types of mistakes will lead to slightly different processing as we will want to yield the drill that will alleviate the root cause mistake. To do so we must check if the mistakes the root cause can lead to were identified as mistakes in the analysis and fill their fix field to link to the drill generated for the root cause. Next we must set the “isProcessed” field to true to avoid overwriting the drill generation later in the process. Finally, we can process the root cause mistake in the same manner as processing a solo mistake.



[Figure 20: Root cause node from the feedback architecture tree]

5.2.1 Front-end Integration

In the previous sections we discussed how the video inputted by the user is analysed by the application and a drill is generated to help alleviate any mistakes that may have occurred. In this section we will discuss how we used the data generated in the processing to provide visual feedback to the user, in order to fully understand what is happening in their swing.

The homepage

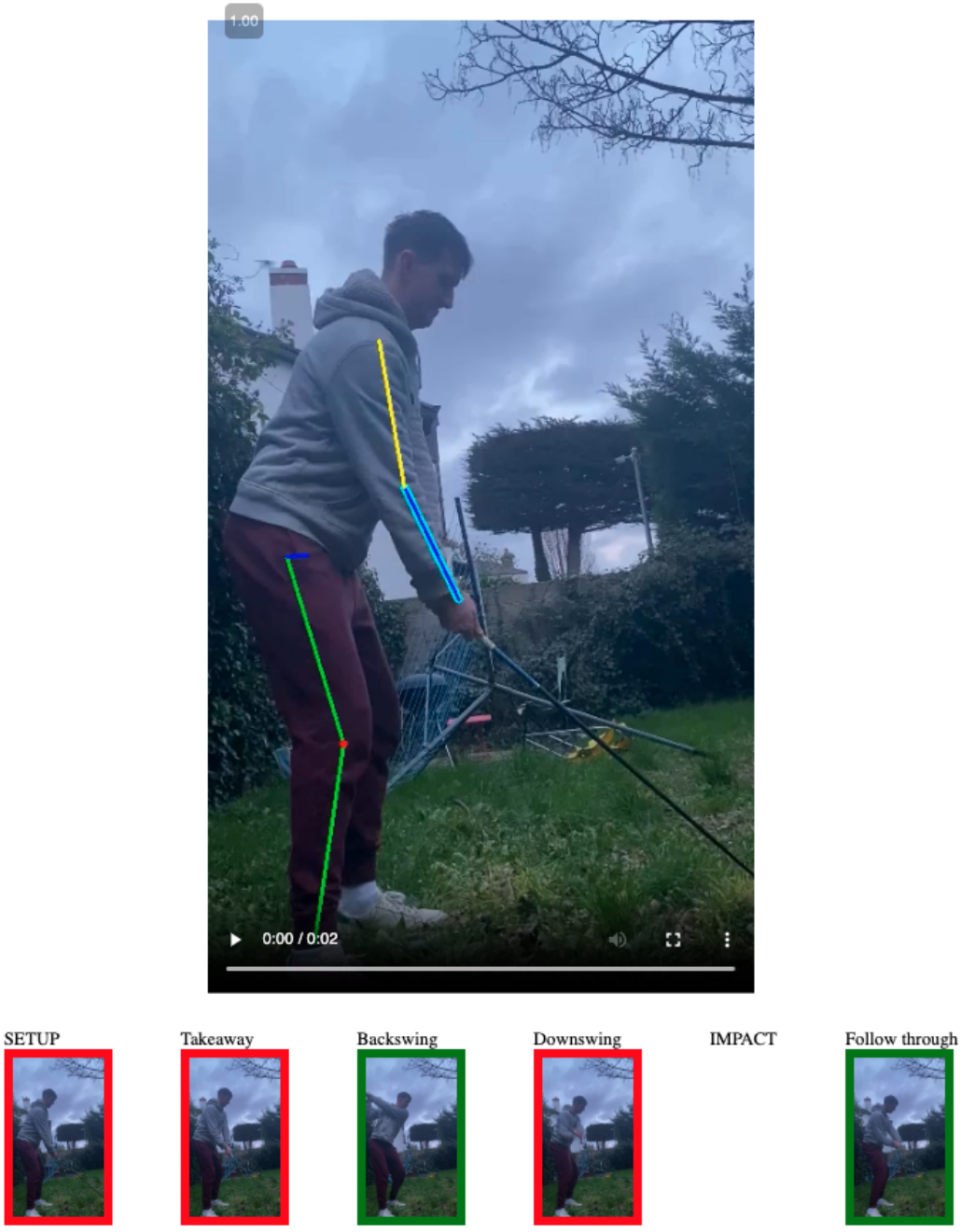
The homepage must provide functionality for the user to upload the video they wish to process. To provide this functionality, we implemented two distinct drag and drop boxes, each representing the view of the video (Face-On or Down-the-LIne). Each of these drag and drop boxes leverages HTML form elements. Forms allow a user to

submit information and send a request to the server with the data stored in the form. Once the video(s) has been submitted to the form, the submit button will upload the video(s) to the server using Javascript. Once the submit button is clicked it will run the UploadVideo script, which will send a POST request to the upload-video endpoint with the video data as the content of the request. Once the server receives this request, the form data will be extracted from the parameters and the video will be saved under the name of the view it represents.

[Figure 21: Home page of the website, displaying drag and drop functionality for file upload

The Analysis Page

The analysis page is split up to 2 endpoints, both use the same template for visualising data however on the server-side they call different functions for the different views. As such, we will be discussing the process for the “Face-On” view, but the process is identical for the face on view, except the methods call for the corresponding view. Once the endpoint for the Face-On analysis page has been reached, the web-server will read the video labelled “Face-On” for processing. The video is first inputted into the video preprocessing function to parse out the irrelevant frames. The parsed video is then classified into the swing-stage frames. These images are saved to the server for further processing. Next the images are fed into the swing analysis module, which will convert the video-data into readable data. Finally this readable data is passed through the feedback module and a drill is generated. The generated readable data is sent to the analysis page from Flask by converting the JSON object of all the checks made to a type that is readable by HTML and JS. The classified swing stage images are also sent to the user. In the analysis page, the 6 stages of the golf swing are displayed. The image is outlined either red or green depending on if a mistake was identified in the analysis for that stage (green represents no mistake identified, red represents a mistake was identified).



[Figure 22: View of the analysis page, displaying red borders around placeholder stage image, to highlight a mistake made]

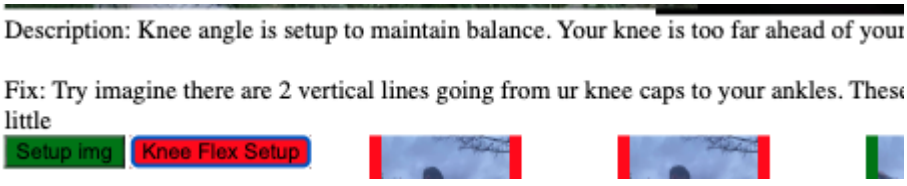
Each of these images act as a placeholder for the checks that were executed at each stage. If an image is hovered over a modal window is opened. The modal contains the image of the swing stage and contains multiple buttons corresponding to a different check that was processed. Again, each of these buttons is coloured either red or green depending on if the check failed or passed.

[Figure 23: Example of the pop up modal when a stage is selected for analysis]

When a button is clicked, the analysis and feedback data is queried for the check that has been requested. As mentioned previously, the feedback object is a JSON array containing all the data required for visualisation. Once the array has been queried, we will have the appropriate check to visualise and give feedback on. The “Points” field of the JSON object contains all the required information to draw the check that was verified. This is done by plotting the points on top of the placeholder image and drawing the appropriate lines/angles.

[Figure 24: Example of points being drawn on the image to verify a check]

Finally for each check that has been selected a description of what was identified in the analysis and the drill generated by the feedback system is outputted. This gives the user information about what exactly occurred in their swing, why it could affect the outcoming shot and a drill to help alleviate this mistake.



[Figure 25: Example of what drill and description is generated when a mistake is processed]

The sections above discussed our implementation by first preprocessing the inputted video, identifying the frames of the video we need to analyse, doing the analysis and finally generating the feedback. This functionality was integrated into an easy-to-use indicative front-end application to provide a visual representation of what had been identified in the user's swing.

6. Evaluation and Results

In the previous sections we have discussed the design and processing of our application. In this section we will discuss the results of the application as a whole. To begin, we must evaluate the methods used in processing the data as this will be important in our results to determine whether or not the application performs as expected in various scenarios. If our program is not robust to variations in the swings of its users it will ultimately fail to make improving more accessible.

6.1 Gathering data

In order to test the results of the application we must gather data of various golf swing videos and information about the resulting shot. The data regarding the shot contains the total length of shot and how far to the side the shot went. With this data and the video of the swing itself, we can determine what went wrong in the shot or if it was good or not. With this data, we will be able to determine if the output data from the application is accurate.

The data contains videos of the author of this paper hitting golf balls. Using only 1 player's swing would restrict the level of experience of the player, so we ensured to practise with different swings, clubs and shot types. As such we have mimicked the swings of different player levels and different swings that have errors in them. The data was processed to isolate the information about the shot and save it in a CSV file, with an associated video name. This data would be used to determine which swing videos should have many mistakes identified and which should have less.

Unfortunately, due to the video data being one person, we have ultimately failed to evaluate the performance on various body types. As such we cannot comment on the performance of the application on different ages, ethnicities or genders. However, as a core model used in this application is MediaPipe BlazePose, our performance on different people is likely similar to that of BazePose.

6.2 Evaluation of methods

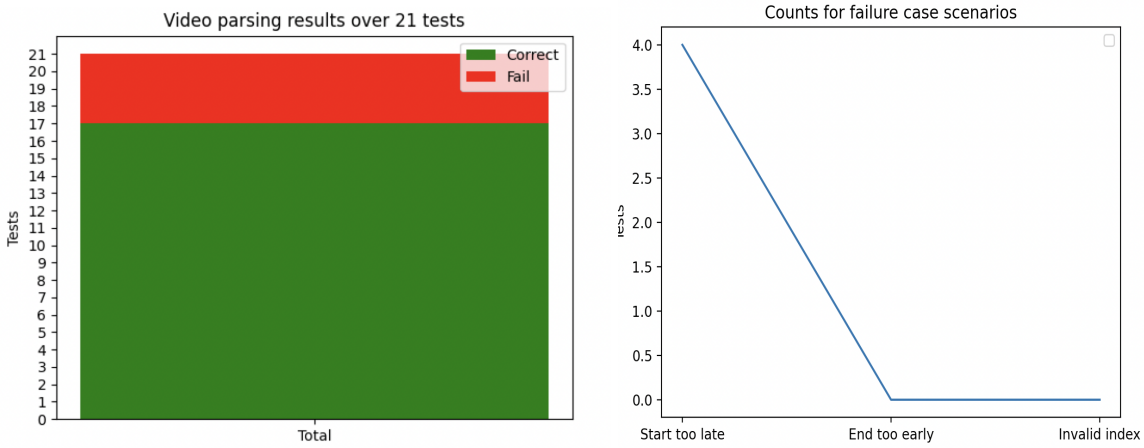
In order to determine whether or not the application was successful in helping golfers, we need to first evaluate the performance of each of our main processes. This is important as this will determine if our implementation works on different

swings and shot types. Unfortunately, we failed to gather enough validation data to properly assess the application. So this area would need further work in order to properly assess the feasibility of this exact implementation. However, we did do enough testing in order to prove these methods as a proof-of-concept to answer the research question given to us.

Evaluation of the video preprocessing steps

If our video preprocessing is not sufficient, it will cause major faults in the application. If this stage fails to include the entire swing movement in the parsed video, the swing stage classification module will fail and so we will be unable to perform further analysis. As a result of the requirement for the swing to be in the parsed video, our evaluation is a simple classifier system. Testing is done by verifying if the swing is in the outputted video. In order to evaluate the performance, we have created our testing framework to return true if the full swing is in the parsed video (at any stage) otherwise false (if the parsed video does not contain the full movement of the golf swing). We manually marked the index of the frames at the beginning and end of the swing movement and verified that these frames exist within the start and end index of the parsed video. Using the manually marked ranges, we verified that the indices for the parsed video fall outside.

Evaluating the video preprocessing stage results in an accuracy of 81%. The left-side graph below shows the performance over 21 videos of various swing styles and clubs and also varying lengths of video.

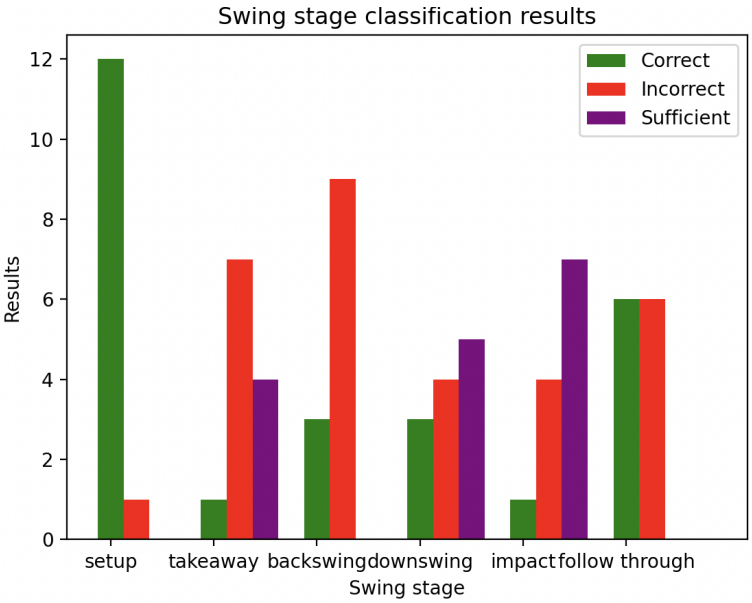


[Figure 26: Displaying the results of the video preprocessing steps]

As we can see, 1 in 5 videos failed to correctly parse. We have done further analysis to see what improvements could be made. The right-side graph displays the total number of failures in each failure case. The main failure case of the system is when the beginning frame index is after the initial movement of the swing, meaning the parser after the initial movement. This shows that our assumption of the impact occurring exactly half-way through the video is untrue. Another failure case is when the highest speed of movement is before the 40th frame of the video, when the program tries to parse the video it gets a corrupt video as the frame index it is trying to retrieve will be a negative value. The final failure case is when the end frame index is calculated too early (i.e. the swing isn't over), however this is very uncommon and we have yet to encounter it in our experimentation, as our assumption that impact is halfway through the swing is untrue, realistically it occurs at the ¾ mark. If we changed the logic, from travelling 40 frames before and after to travel along the data to the next point of minimal movement, (marking the end of follow through) and travelling backwards from impact to the second last period of minimal movement (marking setup), we would see a major boost in accuracy.

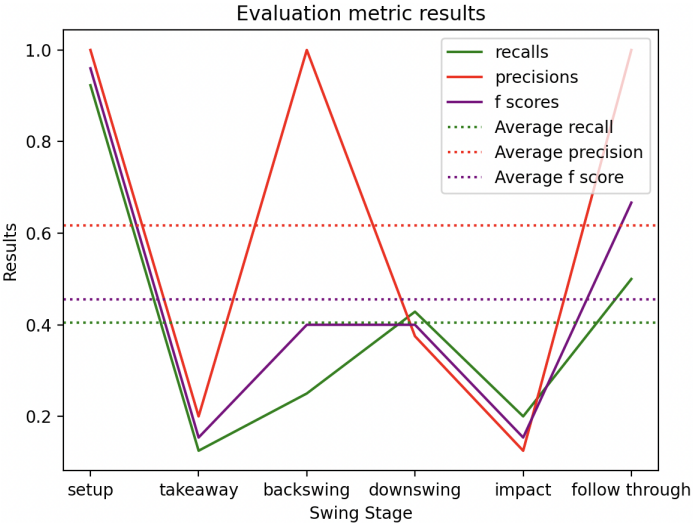
Evaluation of the Swing-Stage Classification module.

We need our swing-stage classifier to correctly estimate most of the stages in every video where possible as we will need to analyse each of these frames. As we failed to find a way to automate this test, we decided to opt for a manual verification program. Whereby, all test videos are processed and the user will be given the output and told to mark it. As multiple of the swing stages can take place over a range of frames (i.e. the backswing is a continuous movement rather than an individual frame), we have marked them with correct, sufficient and bad. Whereby sufficient means it's not a perfect estimation, but it can still be used to verify the checks required. This yields the below results.



[Figure 27: Bar chart displaying performance of the preprocessing methods]

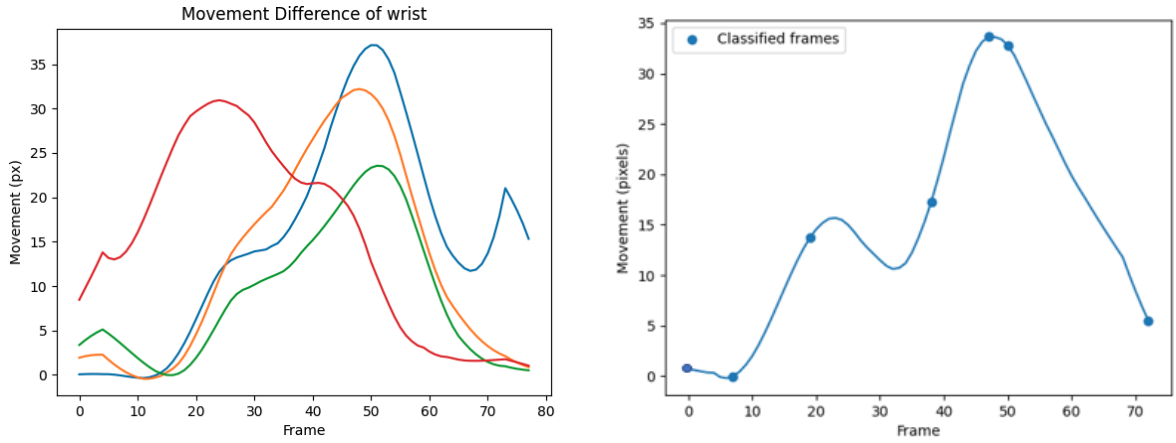
As we can see from these results the performance of the swing stage classification is poor. While most of the results are good enough to use for the checks, the early miscalculations (in takeaway and backswing) lead to further misclassifications. This is as a result of the logic we implemented processing the movement data in both chronological and logical order in the swing, as such a mistake early on has a huge impact as the swing progresses. To further evaluate the performance we need to calculate the precision. Whereby, true positive will be correct, false negative will be incorrect and false positive will be the “sufficient” results.



[Figure 28: Showing the evaluation of the swing stage classification module]

As we can see the results fall drastically in the takeaway which leads to incorrect estimation through the following phases. We also see a dip in the impact frame,

which is expected as the impact frame range of acceptance is far more strict than other stages. From further analysis, we can see the root case of this inconsistency in the processing is the acceleration smoothing functions. As previously mentioned we need the graph to follow an expected shape (as the clubhead should logically speedup), however in practice we do not see this expected shape occur in every instance. Below (left) is an example of various acceleration curves for inaccurate estimations, in comparison to the expected shape (right).



[Figure 29: The graphs on the left, show the acceleration curves of videos that were misclassified. The right side graph, highlight the expected shape for comparison]

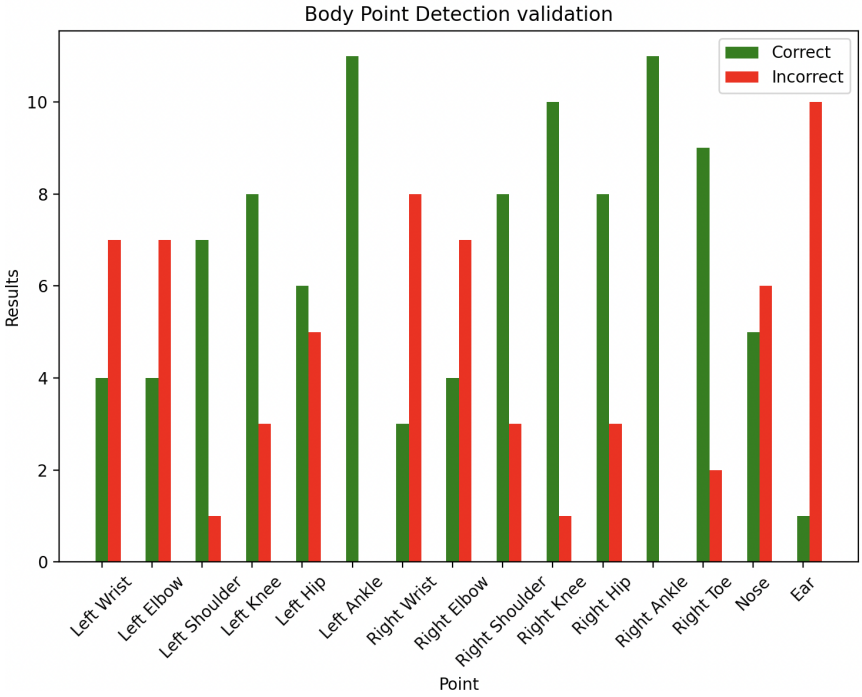
Currently the smoothing of the curve does not guarantee the shape to follow the expected shape as used in the logic of stage estimation. If we changed this to fit a curve of expected parameters (i.e. expected degree of polynomial), we would likely see a major increase in accuracy as we would now guarantee following the appropriate shape. This would completely eliminate the drastically incorrect classifications (e.g. backswing is often classified as follow through due to the closest dip being forward rather than the expected backwards).

Evaluation of Swing Analysis methods

As our swing analysis methods use MediaPipe Pose estimation as the basis of their functionality, we decided to opt against testing the entire pose estimation module and instead opted to test for it in our use-case. In order to do so, we manually marked the points that are required for the checks in 10 frames across a video of a swing. This was done for 16 points, leading to a total of 160 test points. This test method is completely detached from the previous methods. This was chosen, as it leads to

more robust testing (the frames tested include additional frames throughout the range of the movement of the stages as opposed to one general stage frame).

The test data was manually marked, whereby a colour is assigned to the general area of a body point mark on the frame. Then the code verified that the coordinate returned of the actual image corresponds to the correct colour assigned to the appropriate point. Running the tests yields the following results.



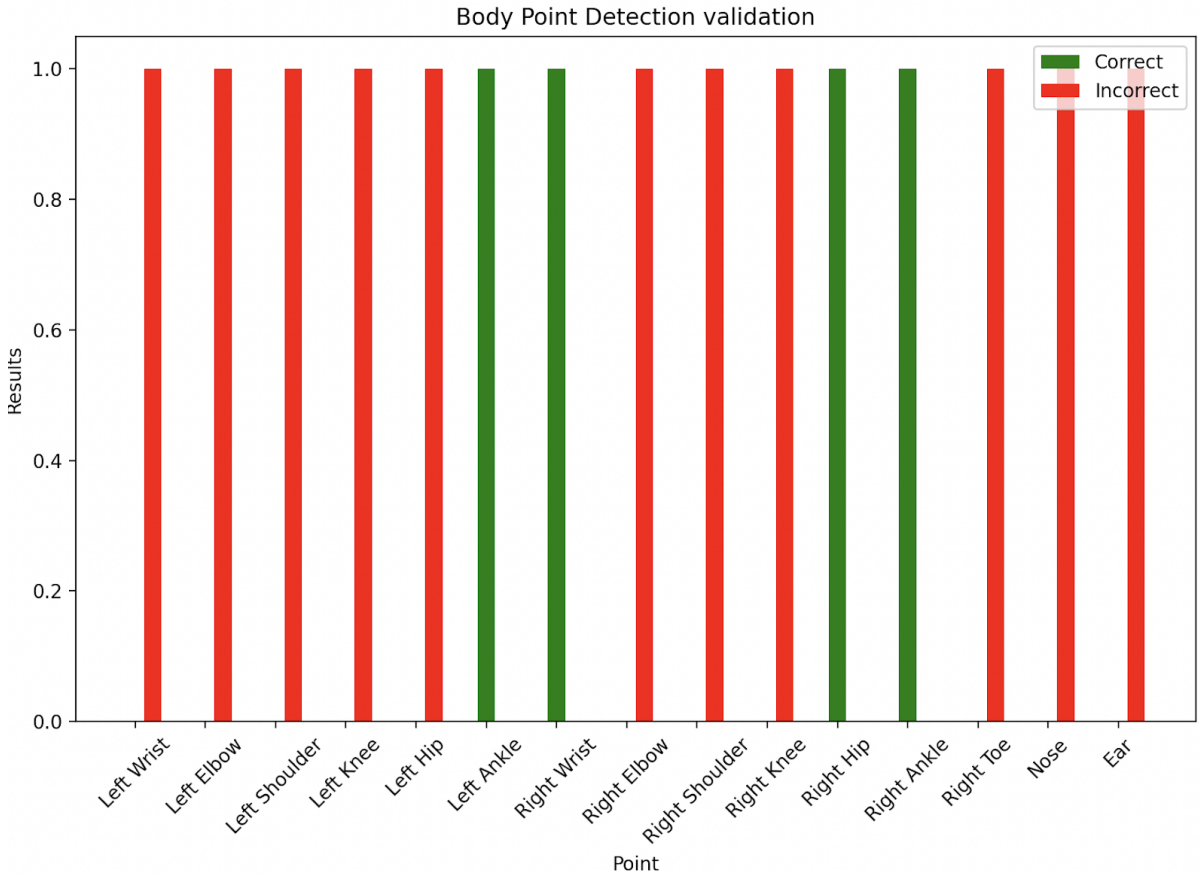
[Figure 30: Graph displaying the performance of the pose estimation in swing analysis]

Calculating the average recall for the model we see a result of 0.61 (61% accuracy). From these results we can see that the smaller points that are often difficult to identify even for a human in the videos (most notably the ear) have a low correct rate. However, the most inconvenient tracking points are both wrists which results are quite poor. This is likely due to the occlusion in the backswing and distortion as the player accelerates for impact. Performing further testing we established that the recall in the 14th frame (takeaway frame, so limited movement and no occlusion) is 0.93, whereas performing it for the 49th frame (point after impact and so most movement and distortion) we see this drop to 0.66. We see the most significant decrease in performance of the model is when the body comes to a standstill (in the backswing and follow through) as it has reached the final point of rotating and is now

rotated away from the camera and so we have occlusion of limbs (examples below).

[Figure 31: Frames displaying occlusion of limbs in the swing as rotation occurs]

In the case of the 63rd frame (left above) we see our recall drop to only 0.26. If we run our graphing on this frame alone we see the below result.



[Figure 32: Showing the performance of the 63rd frame where most occlusion is encountered]

As we can see the correctly detected body marks are clearly visible in the frame image and so are correctly identified. However everything else (with the exception of the right toe and nose. *Note: Ear is referring to the left ear which is occluded in the image*) is incorrectly identified as they are all occluded in the frame image (or at least difficult to exactly locate). However upon further inspection, the detections are rough approximations which are accurate enough to carry out the checks we need to do (example below to verify that the lead leg is roughly below the head). The code is estimating where the left ear is. While the estimate is wrong, it does give the approximated x coordinate of the left ear which is all we require).

[Figure 33: Displaying the rough approximation of body-part location]

Improving the accuracy of this module is difficult as it would require a complete redesign of a pose estimation system. However, a pose estimation system that is specifically designed for golf related movement would be beneficial not only to this application, but to many other developers who are aiming to create programs that analyse a golf swing. An adequate implementation of such an approach would take the general shape of a golf swing into account, track the location for interpretation based on previous detections and consider the speed of change.

6.3 Results

In the previous section we analysed the evaluation of the computer vision methods we used in the implementation of our application. In this section we will discuss the results of the processing. To do so, we will need to determine whether or not the application works in the cases where processing has gone as expected and yields a drill that leads to improvement. As such, due to the inconsistencies in processing we were unable to test the application in practice, however, in order to discuss results we will use various videos and discuss the relevant outcomes.

In order to determine if the application is giving relevant feedback, we use the data about the resulting shot (distance and amount the ball strayed off-course). If we see a shot that is good (far and straight) we know the application should not identify many mistakes. However if we see a shot that is bad (far left or right), we know the program should identify many problems in the swing. To determine this, we use knowledge of the swing itself to further assess the application. As we know if a shot went to the side, it is likely due to the balance of the player being off (as such the checks for balance should fail). We see this behaviour in all the cases that we manually verified in accordance with the shot data. We also see for the cases where the shots are good (long and straight) that few mistakes are identified in the code (provided the swing stage classification executed appropriately). For the shots that we noticed have a lower distance than they should, we know there is a lack of power in the shot and such the rotation axis is likely incorrect. While our application does not necessarily fail in this case, it does struggle to identify the points required to verify the checks that lead to power as many of these are occluded in the rotation of the backswing and downswing. From this assessment we can determine that the application does perform as expected in the majority of cases, provided the swing stage classification occurs as expected. While, it will still fall behind in some of the checks that require estimating a location of a body point, the estimation is usually in a general area and is estimated with a relative degree of accuracy to perform the checks necessary; however this is not guaranteed in all cases.

In order to determine the effectiveness of the feedback given we would need to improve the application to process effectively in every case, as such we have failed

to determine the actual effectiveness of the application in practice. However, to assess the feedback system, the author discussed the feedback with a scratch golfer (a skilled and proficient golfer, who plays to the expected score of a course). In an in-depth conversation discussing the feedback system, changes were made to the responses in order to lead to more effective feedback. However, it was brought to our attention that all our checks are relevant and should be performed for beginners. If we wanted to help amateurs or more advanced players, there is another system that should be analysed in swing assessment. Instead of breaking up the swing into 6 stages, there is a well-known P Classification system[13], that consists of 10 key stages of a golf swing. This system is used for more in-depth analysis and publications have been made of key factors to assess in this system, as such our application currently is directed more-so towards beginners and amateurs as our scope defined. In conclusion, while our results have displayed the basis of an application that will ultimately aid golfers in improving, it currently lacks an adequate implementation of the methods to create such an application. Although, once an adequate implementation has been developed, it is likely that an expansion to help more advanced golfers is possible by integrating the P Classification system and more specific checks.

In this section we have discussed the evaluation of methods used in the computer vision processing of the swing video. We determined the success' and failures of the video preprocessing steps. We analysed the performance of our swing stage classifier and determined the main root of failure causing inaccuracies in classification when various videos with different swing styles are uploaded for analysis. Finally, we evaluated the performance of our pose estimation methods and their feasibility in such a niche application. We discussed improvements that could be made to the models in order to lead to major improvement and a more consistent and effective program. Once the methods had been evaluated, we moved on to discuss the results. We determined the results of our application to be helpful in the majority of cases, however, the miscalculations in classification cause the feasibility of the current application in practice to be sub-par. We also discussed how our application could be expanded, not only to improve the skill-level outlined in the scope of this project, but to expand it to more advanced skill levels.

7. Conclusion

7.1 Further Studies

In this section we will discuss further studies that have been found as an area that has been lacking research. Some of this research will be general research (i.e. no focus on a specific application) and we will describe how it would benefit our methods, other research may be specific to our application.

General Research

In this section we will discuss the research studied with a general use-case. These use-cases are not specific to our application but do have a use in our application.

The first item to discuss is the lack of research in advice feedback systems. Implementing the feedback system in our application was quite tough as we had no experience in feedback systems. This was exacerbated as “little research has been done on classifying advice systems with consideration for human behaviour characteristics”[14]. In trainer based applications, the feedback system is arguably the most important aspect of the application. With an insufficient feedback system the user will fail to improve as they wont have the required knowledge to do so. While computer feedback systems have been studied in abundance, these systems more so deal with feedback loops in machine learning. In applications designed with training systems in mind, an adequate feedback mechanism is necessary. Study should go into designing such a system that produces both advice and visual feedback (i.e. suggest a drill to improve and then generate a video of how this drill is executed). With the boom of generative AI such a feedback mechanism could be integrated by creating a natural language processing system that is specific to the system target (in our case golf) terminology and logic to generate drills that would help the user improve. However, to do such research a lot of prior research will have to be done to understand how these AI systems “understand” what they are saying (i.e. in our case, we would want the AI to understand why the drill suggested will help the user improve). An added benefit of such a system would be iterative feedback. In current advice systems feedback generally follows a tree-like structure and eventually leads to a static node to produce data. As such, by integrating a

generative AI for feedback, the drill produced would likely be different each time and if the user isn't improving this can be taken into account.
The video feedback data could be generated by an AI understanding the visual movement of the golf swing and understanding what drills can influence the movement of certain sections of the swing. This would result in a more personalised and tailored application for each user's golf swing, without the added effort of tailoring. However, the system would have to be able to adjust as the user applies the changes suggested to their swing and the input feed is now seeing a different swing style.

Another area of research that could lead to major benefits is a pose estimation model that can withstand occlusion and high speed movement. As previously mentioned one of the areas our application fell down on was its use of BlazePose, while BlazePose is a sufficient model. Abundant research has gone into general pose estimation models, however limited research has gone into more specific models i.e. when we see high speed movement or occlusion. This research could begin by applying the pose estimation to the movement of a golf swing, as such the model knows the general shape and rate of change of each identified limb, and so when a limb is occluded it could sufficiently interpret the location. This could be done by analysing the previous locations and using the direction of the golf swing and change characteristics seen in previous frames to accurately estimate the location. While this task is even difficult for humans to do (i.e. it is difficult to locate the occluded ear in figure 31), a computer with sufficient data should be able to accurately calculate this to a reasonable degree. Once the model has been expanded to multi-use purpose high speed movement (not specific golf-swing movement and instead specific to all swing-like movements), training applications for these sports could drastically improve by implementing the new model. The model could also be expanded for general high speed movements (i.e. a user on a back cycling past a camera), which would have its own use cases for various applications.

The final area of general research that needs to be done is one that has been studied for decades; however, it is an issue that we dealt with in this application. This research is synchronisation of cameras for low-end devices. While methods of synchronising videos from various angles do exist, these are computationally

expensive and are often adopted for high-cost use-cases. In our initial application design we wished to have an option to process both views in synchronisation with each other. However, due to the lack of ability to synchronise the videos, we were unable to do so. We tried various methods of synchronisation ourselves (getting the same frame in either video and tracking backwards and forwards or using time data in the videos to synchronise them) however, these all failed to give us the degree of synchronisation we required when the frame rates were different. Finding a low-power answer to the synchronisation problem, would have a major benefit in many applications with multiple cameras.

Research specific to our application

In the previous section we discussed previous research that could benefit not only our application but also many other general use-cases. In this section we will discuss the changes to this project that could lead to easier improvements for the user.

The first change to discuss is integrating the P-Classification swing stages into the application. This would be a rather large change, as it would require changes to the classification system and also to the swing analysis stage (as we would now be processing more frames and more checks). This change would include all 10 stages in the system each with their own individual set of fundamental rules. Integrating this change would expand the scope of the project as it would now be focussed on advanced users and cover the whole range of skill-levels. Not only would it expand the scope, it would also introduce a level of granularity, as the user could indicate the level of analysis they want.

The current system implements methods to analyse individual frames of the golf swing, this leaves out numerous mistakes that may have been made and corrected before the next frame is analysed. By analysing the entire video feed, we could identify trends in the golfers movements that are not detected in numerous frames at key stages. This could lead to more in-depth analysis as these trends could be analysed to see their consistency and resulting outcome when they occur and to what degree they occur. An example of such a trend would be the body rotation occurring too early. In our current system we have a frame from takeaway and backswing, however, in the takeaway there should be no rotation and in the

backswing we should be at the end of backward rotation. As such, we have no data about the timing of this rotation or if the rotation is complete at the top of the backswing. By acquiring this data, we could suggest more effective drills to help the user improve.

The final improvement that could be made is to adopt a hybrid approach, whereby some of the processing is done in real time and others after the swing movement. This could be done by analysing the setup and takeaway in real time and giving auditory feedback to the user if a mistake is identified. The user could then adjust their stance to combat this mistake and in turn produce a better shot. However, this real-time processing could only occur for these two stages, as any later than that and any distractions to the user will be counterproductive as the swing may have changed due to their subconscious readjusting due to indication. As such, the remainder of the processing would be done after the swing movement and visual feedback produced for all stages including the setup and takeaway.

7.2 Our Closing Remarks

With computer vision applications gathering more traction in the sporting industry with the introduction of new methods year after year, we wanted to display the potential of creating computer vision based training applications for various sports. In this project we did so by answering the research question “By extracting key-features of a user's golfer swing via computer vision methods, can intuitive feedback be provided to facilitate easier improvements?” within the scope of beginners and amateurs golfers. To answer such a question we had to create an application that uses computer vision to analyse a golf swing.

We began this report by outlining the necessary background knowledge required in order to understand the methodology used in the design of the application. We outlined the theory of the golf swing and the swing characteristics that we would use in order to make the design of the application possible. We defined the key factors that contribute to the success of a golf shot, we did so by discussing what each stage of the golf swing is responsible for. We proceeded by outlining the limitations of our application, which would lay the fundamental and functional boundaries for our

project. The limitations and challenges we faced were the speed of the golf swing leading to distortion in the video feed and also the speed of the clubhead which is too fast to do any meaningful analysis/tracking on. In order to design an application we had to outline the work done in previous literature. We discussed current implementations of computer vision technology in various professional sporting settings (including golf) and also discussed why they are beneficial for their respective sport. We proceeded to discuss similar applications and outline the difference between our proposed product and their current implementations. To highlight the potential of computer vision based training applications we discussed past research into various other sports and how they were similar to what we proposed in this project. This research even included an application that was designed to help Sri Lankan athletes compete in the Olympics. Finally we discussed current research into the golfing field and various papers that would help make our application possible.

In order to create an application that would ultimately help beginner and amateur golfers improve their game, we had to discuss the key factors that have a part to play in the outcome of a golf shot. When outlining the checks we would be performing we referred back to our prior description of the relevancy of each swing stage and described why they were relevant items to verify. The next step in the process was to design the system that would make such an application possible. In the design section we outlined the methodology we would be following in order to create the application. We discussed why we would first preprocess the video to save computation power, then classify the video into images of the swing stages. Once we had images for each of the stages, we explained how we would use Media Pipe Pose Estimation to verify the checks that would be executed to analyse the given swing. Next we discussed how we would analyse the video to generate readable data regarding the swing. Finally, we gave insight into how we would process this generated readable data, to output data regarding drills the user could practise in order to alleviate their mistake. An application with no front-end is not an accessible application, as such the final item we explained was the design of the front end. We discussed how we would design the front end in order to provide intuitive and visual feedback in order to indicate to the user what has gone wrong.

The next section discussed how we implemented the system. We highlighted how we used the movement of the player's wrist between frames in order to parse the video to isolate the swing. Once the swing had been isolated from the video we classified the swing stages. To do this, we had to be more precise with the movement data, as such we discussed the methods we used to preprocess the raw wrist location data and then generate the acceleration curves we would use in logic. The next step in the process was to analyse the swing stage images, we explained how we used pose estimation in order to generate readable data that would be used in the feedback processing stage. In order to generate feedback, we implemented our system to follow a tree structure in order to yield a hardcoded feedback drill.

Finally, we discussed the methods we used in order to evaluate the methods we designed. We explained the methodology behind our test framework and how we implemented them. These main weaknesses were inconsistencies in variations of wrist movement used in swing stage classification and also the poor performance of MediaPipes BlazePose model for high speed movements. The last thing we had to discuss was the results of the application when put to use. Unfortunately due to the inconsistencies mentioned, we were unable to properly assess the application in practice. However, we did discuss how we used prior knowledge of golf in order to assess the results in theory. We established that the application was successful, however, further work is needed to make it feasible to use in real life application. We also outlined the changes that could be made in order to create this functional application, an application with such functionality could drastically change the process of improving in golf. In most cases, golfers have to spend countless hours in expensive lessons with their course professional in order to improve. Whereas, this application could make improving more accessible to any golfer regardless of skill level and could do this free of charge. Not only could this application be used individually, it could also be used by a coach to indicate to them what has gone wrong in the swing and the suggested feedback could be altered by the coach to suit the specific swing of their client.

References

[1] - Vayanos, T. (no date) *Copyright 2018 ©Troy Vayanos*, *SquareSpace*. Available at:

https://static1.squarespace.com/static/59a94d22bebafb3c52bf8f59/t/5b76e26f4fa51a152c63669a/1534517880863/7+steps+to+your+perfect+golf+swing+-+troy+vayanos+-+print+friendly+version-ilovepdf-compressed.pdf

[2] - Kaspriske, R. (2021) *Swing sequence: Tiger Woods*, *Golf Digest*. GolfDigest. Available at: https://www.golfdigest.com/story/swing-sequence-tiger-woods

[3] Tour, P.G.A. (2022) *PGA Tour expands Trackman offering to cover every shot hit - news*, *Irish Golf Desk*. Irish Golf Desk. Available at: https://www.irishgolfdesk.com/news-files/2022/2/2/pga-tour-expands-trackman-offering-to-cover-every-shot-hit

[4] - Thursday, A.13 and Learning, C.V.D.L.M.P.O.-device (no date) *On-device, real-time body pose tracking with MediaPipe Blazepose*, – *Google AI Blog*. Available at: https://ai.googleblog.com/2020/08/on-device-real-time-body-pose-tracking.html

[5] - V. Singh, A. Patade, G. Pawar and D. Hadsul, "trAIner - An AI Fitness Coach Solution," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-4, doi: 10.1109/I2CT54291.2022.9824511

[6] - Chen, Steven and Yang, Richard. (2018). Pose Trainer: Correcting Exercise Posture using Pose Estimation

[7] - H. L. K. Silva, S. D. Uthuranga, B. Shiyamala, W. C. M. Kumarasiri, H. B. Walisundara and G. T. I. Karunarathne, "A Trainer System for Air Rifle/Pistol Shooting," 2009 Second International Conference on Machine Vision, Dubai, United Arab Emirates, 2009, pp. 236-241, doi: 10.1109/ICMV.2009.74

[8] - Meister, D.W. *et al.* (2011) “Rotational biomechanics of the Elite Golf Swing: Benchmarks for amateurs,” *Journal of Applied Biomechanics*, 27(3), pp. 242–251. Available at: https://doi.org/10.1123/jab.27.3.242

[9] - K. Kumada, Y. Usui and K. Kondo, "Golf swing tracking and evaluation using Kinect sensor and particle filter," 2013 International Symposium on Intelligent Signal Processing and Communication Systems, Naha, Japan, 2013, pp. 698-703, doi: 10.1109/ISPACS.2013.6704639.

[10] - T. T. Kim, M. A. Zohdy and M. P. Barker, "Applying Pose Estimation to Predict Amateur Golf Swing Performance Using Edge Processing," in IEEE Access, vol. 8, pp. 143769-143776, 2020, doi: 10.1109/ACCESS.2020.3014186

[11] - Glazier, Paul & Davids, Keith. (2005). Is there such a thing as a 'perfect' golf swing?. International Society of Biomechanics in Sports' Coaches Information Service. http://www.coachesinfo.com

[12] - Crews, Debbie and Phillip J. Cheetham. "Comparison of Kinematic Sequence Parameters between intermediate and Professional Golfers." (2008)

[13] - *The positions of the Golf Swing (P Classification System)* (no date) *The DIY Golfer*. Available at: https://www.thediygolfer.com/blog/the-positions-of-the-golf-swing-p-classification-system

[14] - Q. Liu, H. Itoh and K. Yoshimura, "On the Design of In-vehicle Advice System," 2006 IEEE International Conference on Systems, Man and Cybernetics, Taipei, Taiwan, 2006, pp. 2511-2516, doi: 10.1109/ICSMC.2006.385241.